uc3m | Universidad Carlos III de Madrid

TESIS DOCTORAL

Advanced Techniques in Scientific Computing: Application to Electromagnetics

Autor:

Adrián Amor Martín

Director y tutor:

Luis Emilio García Castillo

DOCTORADO INTERUNIVERSITARIO DE MULTIMEDIA Y COMUNICACIONES

Leganés, octubre de 2018

PUBLISHED AND SUBMITTED CONTENT

During this Ph.D. dissertation, different contributions have been reported to the literature. The complete list is detailed below.

Journals (3):

- **A. Amor-Martín**, L. E. García-Castillo, and D. García-Doñoro, "Second-Order Nédélec Curl-Conforming Prismatic Element for Computational Electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 10, pp. 4384-4395, 2016. Some results from this reference are included in Chapter 4. The material from this source included in this thesis is not singled out with typographic means and references.

- D. García-Doñoro, S. Ting, **A. Amor-Martín**, and L. E. García-Castillo, "Analysis of Planar Microwave Devices using Higher Order Curl-Conforming Triangular Prismatic Finite Elements," *Microwave and Optical Technology Letters*, vol. 58, no. 8, pp. 1794-1801, 2016.

- **A. Amor-Martín**, I. Martínez-Fernandez, and L. E. García-Castillo, "Posidonia: A Tool for HPC and Remote Scientific Simulations [EM Programmer's Notebook]," *IEEE Antennas and Propagation Magazine*, vol. 57, no. 6, pp. 166-177, 2015.

Conference contributions (14):

- **A. Amor-Martín**, L. E. García-Castillo, and D. García-Doñoro, "Towards a Scalable hp Adaptive Finite Element Code based on a Nonconformal Domain Decomposition Method," in *European Microwave Conference (EuMC), 2018*, EuMA, 2018. Some results from this reference are included in Chapter 6. The material from this source included in this thesis is not singled out with typographic means and references.

- D. García-Doñoro, W. Mei, **A. Amor-Martín**, L. E. García-Castillo, and M. Salazar-Palma, "Electromagnetic Finite Element Solver for HPC Environments using Direct Substructuring Method," in *European Microwave Conference (EuMC), 2018*, EuMA, 2018.

- **A. Amor-Martín**, L. E. García-Castillo, and D. García-Doñoro, "Non-conformal Domain Decomposition Method supporting hp Discretizations," in *14th International Workshop on Finite Elements for Microwave Engineering*, IEEE, 2018.

- **A. Amor-Martín**, L. E. García-Castillo, and D. García-Doñoro, "Higher Order Finite Element Method based on a Non-conformal Domain Decomposition Method," in *Emerging Trends in Applied Mathematics and Mechanics 2018*, 2018.

- J. A. Belloch-Rodriguez, **A. Amor-Martín**, D. García-Doñoro, and L. E. García-Castillo, "Acceleration of a Mesh Truncation Technique for a Finite Element Electromagnetics Code," in *18th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2018*, 2018.

- D. García-Doñoro, **A. Amor-Martín**, and L. E. García-Castillo, "Higher-Order Finite Element Electromagnetics Code for HPC Environments," *Procedia Computer Science*, vol. 108, pp. 818-827, 2017.

- **A. Amor-Martín**, D. García-Doñoro, and L. E. García-Castillo, "A Finite Element Mesh Truncation Technique for Scattering and Radiation Problems in HPC Environments," in *Computing and Electromagnetics International Workshop (CEM), 2017*, pp. 33-34, IEEE, 2017. Some results from this reference are included in Chapter 4. The material from this source included in this thesis is not singled out with typographic means and references.

- **A. Amor-Martín**, D. García-Doñoro, and L. E. García-Castillo, "Analysis of Dispersion Error of Higher-Order Curl-Conforming Prismatic Finite Element," in *Numerical Electromagnetic and Multiphysics Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO), 2017 IEEE MTT-S International Conference on*, pp. 203-205, IEEE, 2017. Some results from this reference are included in Chapter 4. The material from this source

included in this thesis is not singled out with typographic means and references.

- **A. Amor-Martín**, D. García-Doñoro, and L. E. García-Castillo, "On the Design of Higher-Order Curl-Conforming Finite Elements and its Assembly Features," in *Numerical Electromagnetic and Multiphysics Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO), 2017 IEEE MTT-S International Conference on*, pp. 200-202, IEEE, 2017.

- L. García-Castillo, I. Gomez-Revuelto, **A. Amor-Martín**, M. Los, and M. Paszynski, "Algorithm for simultaneous adaptation and time step iterations for the electromagnetic waves propagation and heating of the human head induced by cell phone," *Procedia Computer Science*, vol. 108, pp. 2448-2452, 2017.

- **A. Amor-Martín** and L. E. García-Castillo, "Second-Order Nédélec Curl-Conforming Prism for Finite Element Computations," in *13th International Workshop on Finite Elements for Microwave Engineering*, IEEE, 2016.

- D. García-Doñoro, S. Ting, **A. Amor-Martín**, L. García-Castillo, and M. Salazar-Palma, "Higher Order Finite Element Method Solver for the Analysis of Microwave Devices in Planar Technology," in *Microwave Conference (EuMC), 2016 46th European*, pp. 473-476, IEEE, 2016.

- **A. Amor-Martín**, I. Martínez-Fernandez, and L. E. García-Castillo, "Posidonia: A tool for HPC and Remote Scientific Simulations," in *13th International Workshop on Finite Elements for Microwave Engineering*, IEEE, 2016.

- D. García-Doñoro, **A. Amor-Martín**, L. García-Castillo, M. Salazar-Palma, and T. Sarkar, "HOFEM: Higher Order Finite Element Method Simulator for Antenna Analysis," in *Antenna Measurements & Applications (CAMA), 2016 IEEE Conference on*, pp. 1-4, IEEE, 2016.

A mis abuelos.

We scream and shout till we work it out.
Can we just work it out?

Desde aquí, hoy contemplo el final...

Si no puedes hacerlo perfecto,
hazlo raro.

AGRADECIMIENTOS / ACKNOWLEDGEMENTS

Seis años han pasado desde la última vez que me senté a hacer balance para cerrar una etapa. Y en estos seis años he crecido tanto que necesito dar las gracias a todas las personas que me han hecho la persona que soy ahora.

Primero debo dar las gracias a mi director de tesis, a Luise. Has sido un referente a todos los niveles en estos años, y espero no perder nunca la pasión y el entusiasmo con el que salía de todas las reuniones que he tenido contigo. Daba igual que fuera un viernes por la tarde, un domingo de madrugada o un rato entre vuelos, siempre conseguías encontrar el porqué de las cosas. Espero haber asimilado esa pasión por la ciencia e intentar transmitirla en un futuro. También necesito dar las gracias a Doñoro. Dani, los cimientos de esta tesis se construyeron gracias a tu trabajo previo. Tu ayuda al principio de esta tesis ha sido clave para poder bajar al fango de Fortran. Tampoco olvido los buenos ratos pasados en Macao, donde fui feliz, ni esos días tan agradables que pasé en Hong Kong con tu familia. Espero que sigamos colaborando en los próximos años. Y tampoco me puedo olvidar de Nacho, mi director de PFC y cofundador de ScientApps. Si no me hubieras propuesto un proyecto que me pareció asumible (y nos llevó lejos), no habría tenido la valentía de empezar esta tesis.

Del grupo me gustaría destacar el papel que han tenido Dani y Quique durante estos años. Dani, te agradezco la confianza ciega que tuviste en mí cuando no había FPU, ni dinero, ni papers. Junto con Luise, me hicisteis sentir valorado y eso me empujó a seguir mejorando. Además, con tu ejemplo he aprendido a valorar lo difícil que es tener responsabilidad y, a la vez, tratar de ser buena persona. Y Quique, toledano de adop-

ción, muchísimas gracias por darme ánimos en los peores momentos, por pensar en la vida sin dramas, y por todas las conversaciones que hemos tenido y que me han ayudado a ver con más claridad. Has sido un apoyo en todo momento y eso es algo que no podré olvidar.

También me gustaría agradecer la ayuda que me han dado Sergio Llorente y Alejandro G. Lampérez, fuentes de inagotable sabiduría.

Además, he de agradecer la financiación de esta tesis a la Universidad Carlos III de Madrid desde 2014 a 2015 mediante el programa para personal investigador predoctoral en formación, y al Ministerio de Educación, Cultura y Deporte desde 2015 a 2018 por medio de las ayudas para Formación de Profesorado Universitario.

Apart from my Spanish colleagues, this PhD has given me the opportunity to work with other groups in far places as Macau or the US. I would like to thank Prof. Ting to host me in his group together with Prof. Tam. There I met wonderful people as Tim or X and I have very good memories with them.

Also, I would like to thank Prof. Jin-Fa Lee, whose guidance and technical discussions made me grow as a researcher. I also want to thank Dr. Dongwei Li for his help in the implementation of DDM. Apart from the academy, I would like to mention a little group of friends that I found there: Bugra, Václav, Julia, Luigi, Francisco, Matthias, and Naiara. My first stay was much better when I met Marta, with whom I had dinner so late in the afternoon (or at a normal time for us) several days. Regarding my second stay, I would like to thank Miriam (and his husband, Francisco), who was so helpful when I arrived in Columbus and such a cool girl; Mariona, with whom I had long conversations which helped me at the beginning of my stay; and to my only labmate, Arne. It was cool to share that *crowded* office with you and to find a friend who laughed with my Spanish gestures!

Sin embargo, estas estancias no me han impedido pasar largas temporadas en el 4.2.E01, donde he pasado de ser el recién llegado que se sienta en la mesa del medio a tener un confortable sitio con ventana y estantería. Ese laboratorio no sería nada sin todas las personas que me hacían reír en los momentos de más agobio y que han dejado su marca personal día a día. De Edu aprendí que el rigor intelectual no estaba reñido con una elegancia innata ni con saber disfrutar cuando toca; de Iván, las imitaciones y la forma de ser, tan gallega; de Javi, la pasión por la docencia y la investigación; de Álex, la conexión manchega y el

conocimiento práctico inconmensurable, de buen ingeniero; de Rubén, la bondad y lo agradable que eran siempre sus conversaciones; de Javi, el ser una persona sensacional, querida por todo el mundo, y con una sensatez y una estabilidad mental increíbles; de Gato (madre mía... ¿hace cuántos años éramos compañeros de máster?), el saber recibir el trabajo con una sonrisa y el buen ambiente que era capaz de crear siempre; de Ana, el haber sido compañera de mesa y amiga desde el principio y nunca estar seguro de si la reacción a un comentario iba a ser buena o mala; de Gabri, la amistad (aún no es 2019), el cuidado de sus cutículas y los buenos ratos de teambuilding que pasamos; de Jose, la sensatez y la risita inquisidora que me hacía reír siempre; de Sergio, el preocuparse porque el resto del laboratorio esté bien; de Alberto, la sinceridad y los saludos; de Gabriel, la pasión por la ciencia, el sentido del humor y esas preguntas que siempre me hacen pensar (y la revisión tan detallada que ha hecho de la tesis); y de Kerlos, la constancia en el trabajo, la bondad y nuestra pasión compartida por el fútbol y las palmeras.

No podría olvidarme de todas esas personas que consiguen mantenerme unido a la realidad cuando me costaba desconectar del trabajo. Quiero agradecer todos estos años de amistad a mi grupo de Toledo, esos en los que pienso cuando me pregunto quién organizará mi despedida de soltero. Reven, Mario, Estrella, Tania, Roberto, Ester, Lota, Ali, Ana, Clau... los fines de semana han sido mi vía de escape para poder recargar pilas. Especialmente tengo que dar las gracias a Sergio, por ser parte del núcleo duro toledano siempre disponible para echar un rato; a David, por no cambiar en todos estos años en los que se ha convertido en el doctor Bolonio; a Álex, por tener algo inexplicable que hace que den igual los meses que no nos veamos; a Manu, por haber encontrado su sitio en este mundo y siempre sonreír a la vida; a Samu, por tener esa conexión que nos hace reírnos de completas estupideces; a Bachi, por haber sido mi compañero de referencia tantos años; y a Litos, por su amistad incondicional y querer ser mi padrino. También quiero dar las gracias a todas aquellas personas que he ido conociendo con el paso de los años y me han ayudado a sentirme acompañado aún estando a miles de millas de distancia: a Preeti, por mantener siempre una visión positiva de la vida; a Kuki, por todos los conciertos a los que hemos ido juntos, y por haber sido una persona importante en mi vida a pesar de la distancia; a Lara, por poder vernos y estar hablando de nuestras vidas como si nos viéramos todos los días; a Cris, por haber estado en

todos los momentos importantes de mi vida y ser un apoyo, siempre; y a Leti, con la que da igual el país, el huso horario, la vida... siempre siento que está ahí para escucharme y darme una opinión sincera, para poder hablar por teléfono y hacerme sonreír con un... ¿qué pasa, tú?, para ser el yang.

Por penúltimo, necesito dar las gracias a Lula, sin la que estos últimos meses habría perdido la cabeza. Me siento muy afortunado de haber encontrado una compañera de viaje con la que puedo tener total confianza, sentir que no estamos solos, mirarla a los ojos y ver mis próximos años. También a mis padres, porque nada de esto se construye sin una referencia y un apoyo incondicional a cada una de mis decisiones, incluso aunque éstas supongan irse lejos, a otras ciudades, a otros países. Gracias por no haberos quejado de que la ciencia prácticamente os ha robado un hijo, del que espero que estéis orgullosos. Todo esto es vuestro.

Y, finalmente, queda la parte más difícil y que nunca me gustaría haber escrito. Quiero dar las gracias a mi abuela, por habérmelo hecho todo tan fácil, por haber luchado siempre con buena cara, por haber dado ejemplo sin decir nada. Por haber inculcado en mí, junto con mi abuelo, la cultura del esfuerzo que ha hecho posible esta tesis. Porque aunque no nos lo hayamos dicho apenas, os he querido y os seguiré queriendo con todas mis fuerzas, y espero que os haya hecho sentir queridos al menos la mitad de lo que me he sentido yo estos últimos diez años. Muchas gracias, intentaré estar a vuestra altura. Te echo de menos.

CONTENTS

RESUMEN

Durante los últimos años, los componentes de radiofrecuencia que forman parte de un sistema de comunicaciones necesitan simulaciones cada vez más exigentes desde el punto de vista de recursos computacionales. Para ello, se han desarrollado diferentes técnicas con el método de los elementos finitos (FEM) como la conocida como adaptatividad *hp*, que consiste en estimar el error en el problema electromagnético para generar mallas de elementos adecuadas al problema que obtienen una aproximación de forma más efectiva que las mallas estándar; o métodos de descomposición de dominios (DDM), basado en la división del problema original en problemas más pequeños que se pueden resolver en paralelo. El principal problema de las técnicas de adaptatividad es que ofrecen buenas prestaciones para problemas bidimensionales, mientras que en tres dimensiones el tiempo de generación de las mallas adaptadas es prohibitivo. Por otra parte, DDM se ha utilizado satisfactoriamente para la simulación de problemas eléctricamente muy grandes y de gran complejidad, convirtiéndose en uno de los temas más actuales en la comunidad de electromagnetismo computacional.

El principal objetivo de este trabajo es estudiar la viabilidad de algoritmos escalables (en términos de paralelización) combinando DDM no conformes y adaptatividad automática en tres dimensiones. Esto permitiría la ejecución de algoritmos de adaptatividad independiente en cada subdominio de DDM. En este trabajo se presenta y discute un prototipo que combina técnicas de adaptatividad y DDM, que aún no se han

tratado en detalle en la comunidad científica. Para ello, se implementan tres bloques fundamentales: *i)* funciones de base para los elements finitos que permitan órdenes variables dentro de la misma malla; *ii)* DDM no conforme y sin solapamiento; y *iii)* algoritmos de adaptatividad en tres dimensiones. Estos tres bloques se han implementado satisfactoriamente en un código FEM mediante un método sistemático basado en el método de las soluciones manufacturadas (MMS). Además, se ha llevado a cabo una paralelización a tres niveles: a nivel de algoritmo, con DDM; a nivel de proceso, con MPI (Message Passing Interface); y a nivel de hebra, con OpenMP; todo en un código modular que facilita el mantenimiento y la introducción de nuevas características.

Con respecto al primer bloque fundamental, se ha desarrollado una familia de funciones base con un enfoque sistemático que permite la expansión correcta del espacio de funciones. Por otra parte, se han introducido funciones de base jerárquicas de otros autores (con los que el grupo al que pertenece el autor de la tesis ha colaborado estrechamente en los últimos años) para facilitar la introducción de diferentes órdenes de aproximación en el mismo mallado.

En lo relativo a DDM, se ha realizado un estudio cuantitativo del error generado por las disconformidades en la interfaz entre subdominios, incluidas las discontinuidades generadas por un algoritmo de adaptatividad. Este estudio es fundamental para el correcto funcionamiento de la adaptatividad, y no ha sido evaluado con detalle en la comunidad científica.

Además, se ha desarrollado un algoritmo de adaptatividad con prismas triangulares, haciendo especial énfasis en las peculiaridades debidas a la elección de este elemento. Finalmente, estos tres bloques básicos se han utilizado para desarrollar, y discutir, un prototipo que une las técnicas de adaptatividad y DDM.

ABSTRACT

In the last years, more and more accurate and demanding simulations of radiofrequency components in a system of communications are requested by the community. To address this need, some techniques have been introduced in finite element methods (FEM), such as *hp* adaptivity (which estimates the error in the problem and generates tailored meshes to achieve more accuracy with less unknowns than in the case of uniformly refined meshes) or domain decomposition methods (DDM, consisting of dividing the whole problem into more manageable subdomains which can be solved in parallel). The performance of the adaptivity techniques is good up to two dimensions, whereas for three dimensions the generation time of the adapted meshes may be prohibitive. On the other hand, large scale simulations have been reported with DDM becoming a hot topic in the computational electromagnetics community.

The main objective of this dissertation is to study the viability of scalable (in terms of parallel performance) algorithms combining non-conformal DDM and automatic adaptivity in three dimensions. Specifically, the adaptivity algorithms might be run in each subdomain independently. This combination has not been detailed in the literature and a proof of concept is discussed in this work. Thus, three building blocks must be introduced: *i)* basis functions for the finite elements which support non-uniform approximation orders p; *ii)* non-conformal and non-overlapping DDM; and *iii)* adaptivity algorithms in 3D. In this work, these three building blocks have been successfully introduced in a

FEM code with a systematic procedure based on the method of manufactured solutions (MMS). Moreover, a three-level parallelization (at the algorithm level, with DDM; at the process level, with message passing interface (MPI), and at the thread level, with OpenMP) has been developed using the paradigm of modular programming which eases the software maintenance and the introduction of new features.

Regarding first building block, a family of basis functions which follows a sound mathematical approach to expand the correct space of functions is developed and particularized for triangular prisms. Also, to ease the introduction of different approximation orders in the same mesh, hierarchical basis functions from other authors are used as a black box. With respect to DDM, a thorough study of the error introduced by the non-conformal interfaces between subdomains is required for the adaptivity algorithm. Thus, a quantitative analysis is detailed including non-conformalities generated by independent refinements in neighbor subdomains. This error has not been assessed with detail in the literature and it is a key factor for the adaptivity algorithm to perform properly. An adaptivity algorithm with triangular prisms is also developed and special considerations for the implementation are explained. Finally, on top of these three building blocks, the proof of concept of adaptivity with DDM is discussed.

LIST OF FIGURES

LIST OF TABLES

LIST OF PSEUDOCODES

CHAPTER 0

NOTATION

In this chapter, definitions and notations used throughout this document are introduced. Capital letters denote matrices while boldface letters represent vectors. Unit vectors are distinguished with a hat — e.g. $\hat{\mathbf{n}}$ denotes unit normal vector, always defined outward the element or domain—. Imaginary unit is denoted as j. Regarding differential operators, the gradient is defined as $\nabla\bullet$, the divergence is as $\boldsymbol{\nabla}\cdot\bullet$ and the curl is denoted as $\boldsymbol{\nabla}\times\bullet$. When divergence and curl are applied to a surface, subscript τ is used, such as $\boldsymbol{\nabla}_\tau$. Euclidean norm of a vector $\mathbf{x}$ is defined as $\|\mathbf{x}\|_2$. With respect to electromagnetic magnitudes, electric field is denoted as $\mathbf{E}$ when defined in the whole volume and as $\mathbf{e}$ when defined on a surface; $\mathbf{J}$ is the electric current; wavenumber and wave impedance in free space are defined as k_0 and η_0; ε is the electric permittivity while μ is the magnetic permeability for materials, and when subscript $\bullet_r$ is used, these magnitudes are relative to vacuum. The propagation constant is included here as γ. Frequency and angular frequency are denoted as f and ω respectively. Domain is denoted as Ω while its boundary is $\partial\Omega$.

Regarding domain decomposition methods (DDM), when the subscript i is applied to some magnitude, this magnitude is only defined in the subdomain i. The interface between two neighbor subdomains i and j is denoted as Γ_{ij}. The tangential component of some vector magnitude is obtained through the operator π_τ defined as:

$$\pi_\tau(\bullet) := \hat{\mathbf{n}} \times (\bullet \times \hat{\mathbf{n}}) \tag{1}$$

with $\hat{\mathbf{n}}$ being the outward pointing unit vector on the surface where this

operator is applied. Analogously, the operator $\pi_\tau^\times$ to obtain the twisted tangential component is defined as:

$$\pi_\tau^\times(\bullet) := \hat{\mathbf{n}} \times \bullet \tag{2}$$

When dealing with basis functions, a reference element is used with spatial coordinates denoted as ξ, η and ζ; while coordinates related to real elements are x,y and z. The Nédélec mixed-order space of order k is defined as $\boldsymbol{\mathcal{R}}_{\boldsymbol{k}}$, and polynomial spaces are denoted as $\mathbf{P}_k$. If these spaces are applied to some geometry, T is the reference triangle while I is the reference segment. Basis functions are defined as $\mathbf{N}$, $\mathbf{N}^r$ in the real element and reference element, respectively.

Dirichlet (to impose the value of the unknown field), Neumann (to impose the value of the derivative of the unknown field) and Cauchy (as a hybrid of the former two) boundary conditions are denoted as Γ_D, Γ_N and Γ_C respectively. If these boundary conditions belong to the boundary of the domain, they are denoted as $\partial\Omega_\mathrm{D}$, $\partial\Omega_\mathrm{N}$ and $\partial\Omega_\mathrm{C}$. Regarding spaces of functions, boldface letters are used for vector spaces of functions while capital letters stand for scalar spaces of functions. The space of curl-conforming functions, [7], is:

$$\mathbf{H}(\mathrm{curl},\Omega) := \left\{\mathbf{w} \in [L_2(\Omega)]^3 \middle| \boldsymbol{\nabla} \times \mathbf{w} \in [L_2(\Omega)]^3\right\} \tag{3}$$

where $L_2(\Omega)$ stands for the space of square integrable functions over Ω.

To enforce the tangential continuity of a field and impose (weakly) continuous curls in Ω, Dirichlet as essential boundary condition can be used, so:

$$\mathbf{H}_0(\mathrm{curl},\Omega) := \{\mathbf{w} \in \mathbf{H}(\mathrm{curl},\Omega) | \pi_\tau(\mathbf{w}) = \mathbf{0} \,\mathrm{on}\, \partial\Omega_\mathrm{D}\} \tag{4}$$

and in this space is where the electric field reside with the imposition of Dirichlet boundary conditions on perfect electric conductor (PEC) surfaces. Physically, the definition of this space means that electric and magnetic energies are finite.

For DDM trace spaces have to be defined on the surface; specifically:

$$\mathbf{H}^{-1/2}(\mathrm{curl}_\tau,\partial\Omega) := \left\{\mathbf{w} \in \mathbf{H}^{-1/2}(\partial\Omega) \middle| \boldsymbol{\nabla}_\tau \times \mathbf{w} \in \mathbf{H}^{-1/2}(\partial\Omega)\right\} \tag{5}$$

$$\mathbf{H}^{-1/2}(\mathrm{div}_\tau,\partial\Omega) := \left\{\mathbf{w} \in \mathbf{H}^{-1/2}(\partial\Omega) \middle| \boldsymbol{\nabla}_\tau \cdot \mathbf{w} \in H^{-1/2}(\partial\Omega)\right\} \tag{6}$$

are the spaces of functions obtained when operators $\pi_\tau(\bullet)$ and $\pi_\tau^\times(\bullet)$ are applied to functions which belong to $\mathbf{H}(\mathrm{curl}, \Omega)$, that is:

$$\mathbf{H}^{-1/2}(\mathrm{curl}_\tau, \partial\Omega) := \{\pi_\tau(\mathbf{w}) | \mathbf{w} \in \mathbf{H}(\mathrm{curl}, \Omega)\} \tag{7}$$

$$\mathbf{H}^{-1/2}(\mathrm{div}_\tau, \partial\Omega) := \left\{\pi_\tau^\times(\mathbf{w}) \middle| \mathbf{w} \in \mathbf{H}(\mathrm{curl}, \Omega)\right\} \tag{8}$$

Finally, for scalar variables, Sobolev space $H^{1/2}(\Omega)$ is used, [7].

CHAPTER 1

INTRODUCTION

1.1 Background

Nowadays, a system of communications can be decomposed into different parts: radio frequency (RF), control electronics, and signal processing modules. The design of this system covers a number of different technical fields such as electromagnetism, solid-state physics, data processing techniques and information theory.

Lately, the design of these systems must be more and more accurate due to the increase in the demand of bandwidth in the new communication services. Regarding the RF module, full-wave electromagnetic simulations are mandatory to address these requirements so different techniques have been developed in the last decades, including the method of moments (MoM), finite-difference time-domain (FDTD) and finite element method (FEM), [8, 9]. These techniques are focused on providing a complete solution to Maxwell's equations from different approaches: MoM is based on integral equations through the definition of Green's functions, whereas FDTD uses a differential formulation which discretizes differential equations with central-difference approximations. FEM is based on the definition of small elements where the solution is computed (through a variational formulation obtained, e.g., by Galerkin's methods), and then assembled. The typical formulation of MoM, [10], needs a surface mesh and generates a dense matrix which can be accelerated with methods as the multilevel fast multipole algorithm (MLFMA), [11]; it is commonly used for metallic and layered antennas since no truncation box has to be defined for far-field radiation, but

problems arise when complex geometries or materials (with anisotropies or inhomogeneities) are modeled. On the other hand, FDTD, [12], is highly efficient in terms of computational effort since matrix solutions are not needed and complex materials are easily handled. Moreover, with only one simulation of a time pulse (e.g., a Gaussian pulse) a broadband frequency simulation can be obtained through Fourier transforms. However, geometrically complex or resonant structures are very challenging for this technique due to the rectangular grid commonly used for the discretization of the space.

FEM, [1], has been proven as an accurate and flexible tool in the last years since complex geometries and materials can be handled naturally by its formulation. The research group to which the author belongs has made an impact in the developing of FEM, [13–15], and this line is continued in this dissertation. Main drawbacks of this method are the generation of volumetric meshes and the need, as in FDTD, of defining a truncation box to obtain far-field components, which is mandatory for radar cross section (RCS) and antenna analysis. To overcome these difficulties, more and more robust volume mesh generators have been introduced in the last years, [16–18] while different approaches have been proposed to reduce the truncation box, [15] and references therein, [19].

FEM can be summarized into four steps: *i)* definition of the problem, including electromagnetic features (such as materials and boundary conditions) in the geometry; *ii)* generation of the volume mesh, which will provide smaller elements (the so-called finite elements on which the method is based); *iii)* filling and assembly of the matrices related to each element to build a system of equations which is solved; and, *iv)* approximation of the electric or magnetic field by means of a weighted sum of vector polynomial functions, where the weights or coefficients are obtained through the solution computed in the previous step. Thus, these vector polynomials functions (from now on, called basis functions) play a key role in vector FEM.

In fact, following the classic definition of a finite element by Ciarlet, [20], a finite element is composed not only of its geometrical domain but also of a space of functions (expanded by a set of basis functions) and of a set of degrees of freedom. Vector finite elements (commonly used in computational electromagnetics (CEM)) can be decomposed in terms of the continuity imposed between neighbor elements into curl-conforming and div-conforming elements. Curl-conforming elements provide tan-

gential continuity and leave the normal component to be discontinuous between elements, whereas div-conforming elements impose normal continuity between neighbor elements. Due to the tangential continuity of the electric and magnetic fields at the interface between media, curl-conforming elements are more appropriate for the representation of these magnitudes. Within curl-conforming finite elements, a further distinction can be made depending on the completeness of the space of functions, being mixed-order elements commonly used. Mixed-order elements are differentiated from polynomial complete elements by having a constrained representation of the vector fields with different order of polynomial approximations depending on the direction considered. This kind of elements provides a mixed-order approximation of the field while a full-order $p-1$ representation of the curl of the field is retained. In the literature, main references about curl-conforming mixed-order finite elements are [21, 22], although different approaches have been proposed as discussed in Section 1.2.

Basis functions are a key factor in the solution of the system of equations generated since conditioning of the matrix is determined by the choice of these basis functions, and the solving step of the system of equations is strongly affected by this conditioning, [23]. The order of basis functions also leads to different approximations of the electric field: although a better approximation of the field is obtained with higher orders, the assembly of the finite element matrices is more time-consuming specially from third or fourth orders, [24]. Usually, second-order basis functions are used, [2, 25], although non-uniform orders into the same domain are also possible, [26]. This last option is called p refinement, and it consists of increasing the order p of the basis functions in suitable areas. However, the most common refinement is the so-called h refinement, which means having smaller elements in some areas of the mesh. Finally, hp refinements are obtained when these two refinements are combined, [27–29], and although exponential convergence curves can be obtained through the generation of adapted hp meshes, the time needed to obtain an optimal hp mesh may be prohibitive in three-dimensional problems.

The main characteristic of the system of equations obtained is referred to its population: the resulting FEM matrix is highly sparse so normally direct solvers are used to obtain the coefficients associated to the basis functions. Direct solvers are commonly based on the so-called

LU decomposition, and accuracy up to machine precision is provided. They are highly robust but their main drawback is the memory requirement constituting the limiting factor for large scale simulations, [30, 31], even though strong efforts have been made for parallel versions, such as multifrontal massively parallel solver (MUMPS), [32] and math kernel library (MKL) PARDISO, [33]. Finally, it is worth noting that out-of-core strategies are available but performance is substantially deteriorated.

For this reason, direct solvers are not an option for very large scale electromagnetic simulations. Iterative solvers are the other approach to solve a linear system of equations, where different estimations of the solution are built until a good approximation (defined with some residual) is achieved. An important family within iterative solvers are Krylov solvers, where the solution is inside the so-called Krylov subspace, [34]. The main advantage of this kind of methods is that only matrix-vector and vector-vector products are required, allowing a good scaling on high performance computing (HPC) infrastructures, [35]. However, traditional formulations of FEM when involving wave propagation problems usually show a poor performance with iterative solvers and the main problem becomes to find a good preconditioner to improve the convergence of the solver. One of the most commonly used libraries to introduce this kind of solvers is portable, extensive toolkit for scientific computation (PETSc), [36], where several different iterative solvers are available.

To address this memory problem, different DDM have been introduced in the community in the last years. In short, DDM is based on the division of the original problem into smaller subdomains and, then, applying a two-step procedure to obtain the solution in the full problem: first, unknowns defined on the interface between subdomains are extracted, constituting a smaller global surface problem not as sparse as the original problem. Then, the solution on the surface is propagated to the interior of each subdomain. DDM is an active topic on the field for a number of important reasons:

- DDM is inherently parallelizable since each subdomain can fill and assemble its local matrix, then construct the global surface problem which can be solved in parallel, and finally each subdomain computes the solution from the interface to other subdomains.

- DDM formulation may support non-matching interfaces between

subdomains, which can be used to mesh independently each subdomain (even with different finite element shapes).

- DDM can be used as a preconditioner for iterative solvers.
- DDM can be used to perform full-wave simulations of periodic structures with only one local subdomain matrix.

The first three reasons are addressed in this work and are present in the literature, as it is shown in the detailed review regarding time-harmonic Maxwell's equations included in Section 1.2. This set of advantages lead to perform full-wave simulations that without DDM would not be possible.

On top of that, non-conformal DDM may be used as a technique to obtain *hp* adaptivity scalable codes, in terms of parallelization, if the *hp* refinement is defined locally in each subdomain. This possibility can alleviate the computational effort which makes prohibitive the generation of *hp* refinements in three dimensions. Furthermore, a reduced set of the subdomains obtained with DDM may be repeatedly refined while the remaining subdomains keep the same mesh and calculations associated.

Finally, the FEM implementation shown in this code is derived from the in-house electromagnetic simulator from [37]. In the last decade, this simulator has been written from scratch using modern Fortran constructions and object-oriented programming (OOP) paradigms following a modular organization. This modular programming allows different finite element shapes, basis functions and solvers living simultaneously in the same code, as shown in Figure 1.1. Since this code is in constant evolution, an automatic set of tests based on the method of manufactured solutions (MMS), [13, 38] has been introduced to debug new contributions. This code has been proven as a reliable tool for a number of different electromagnetic problems, [14, 31, 37, 39–41] so when introducing new techniques as DDM or adaptivity in this dissertation no comparison with commercial software or measurements are introduced.

1.2 Literature review

1.2.1 Basis functions

A growing interest in the mathematical and CEM communities has arisen from seminal papers by Nédélec [21, 22], where curl-conforming mixed-order finite elements were established for a number of shapes

(tetrahedra, triangular prisms and hexahedra). These curl-conforming elements overcome the problem of the spurious modes (numerical approximations of non-physical solutions but mathematically valid for the double-curl formulation) through the correct approximation of the spurious solutions. A detailed review can be found in [1,27,42], and remarkable works from the mathematical community are included in [43, 44], including the analysis of commutavity with de-Rham diagrams to prove the sufficient condition for stability. From the electromagnetics community, works from [45–49] have made an impact. Two main families of basis functions can be distinguished: *i)* hierarchical, where a set of shape functions of order p contains all the shape functions of lower orders, [3]; and interpolatory, where basis functions are designed to have non null trace at one entity (edges, faces) and null at the others.

The systematic approach included in [50,51] for tetrahedra has been expanded in this dissertation to triangular prisms, [52]. The space of basis functions proposed in this dissertation is based on a tensor product of the two dimensional simplex (i.e., triangle) and the one dimensional (i.e., segment) spaces. This tensor product has to be defined in the spaces of functions and not the basis functions themselves to preserve stability and unisolvency of the degrees of freedom, [22]. A number of basis functions with no definition of the space of functions a priori are available in the literature for triangular prisms: [53], non curl-conforming in the general case; [54], which use Sylvester polynomials to obtain basis functions of generic order; [55, 56], generated with the imposition of a number of constraints; and [57] which is a spectral version of the triangular prism.

1.2.2 DDM

As commented before, DDM is a hot topic in the FEM community and contributions from different fields have been introduced in the literature, specially from the mathematical and CEM community. First, different concepts related to DDM have to be introduced: *i)* overlapping is referred to the division of subdomains (non-overlapping DDM is obtained when the division is disjoint); *ii)* conformality applied to a surface usually means that the mesh is the same on both sides of the interface between subdomains; however, in this dissertation, conformality is also related to the finite element of each subdomain; and *iii)* matching meshes are relative to the discretization of the mesh on the interface.

In this work, non-overlapping DDM is used while different schemes of conformality and matching interfaces have been tested. E.g., in Fig-

ure 1.2, two different schemes for DDM are shown for illustration purposes:

- Between Ω_1 and Ω_2 a matching interface is present. Although the finite element shapes are different, the division may be conformal or non-conformal depending on the basis functions used for each domain: i.e., if the elements belong to the same family of basis functions and same approximation order p is used, this division is conformal; otherwise, the division is non-conformal.
- Between Ω_2 and Ω_3 a non-matching interface is defined, even although Ω_2 and Ω_3 are discretized with the same finite element.

Regarding the solution of time-harmonic Maxwell's equations, three important families can be distinguished: *i)* optimized Schwarz methods (OSM), mainly developed by the mathematical community which allows overlapping between subdomains; *ii)* cement element method, leaded by Prof. Jin-Fa Lee's group; and *iii)* finite element tearing and interconnecting (FETI) techniques through the definition of Lagrange multipliers whose main contributor is Prof. Jian-Ming Jin's group. It has to be noted that the second family has been introduced in the framework of the first family in [58] and in the machinery of the third family in [59].

Many contributions to the study of the transmission conditions (TC) are introduced by the first family of DDM introduced here. First non-overlapping DDM is proposed in [60], while one of the first contributions regarding the optimization of TC is detailed in [61]. Convergence of the iterative solver is strongly affected by the definition of TC and different optimizations can be found in the literature. Some of these contributions are included in two libraries: HPDDM, [62] built with an interface to FreeFEM++, [63], and GetDDM, [64], which makes use of Gmsh, [18]. Regarding the first library, some contributions are detailed in [65], where optimized restricted additive Schwarz (ORAS) preconditioner is defined, and [66], where a different optimization of the TC is included. This family of DDM can be used in a non-overlapping, [67], or overlapping fashion through the definition of a partition matrix as thoroughly explained in [68]. Interesting results applied to brain microwave imaging are also included in [69, 70]. With respect to GetDDM, contributions from [71–73] are included. In these references DDM is integrated with the on surface radiation condition (OSRC), [74], and finite conductivity in conductor materials is taken into account, [75]. In [72], it is explained

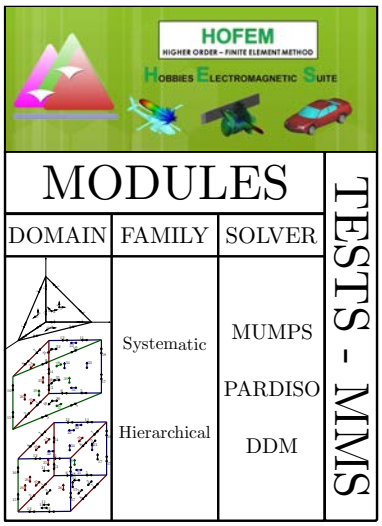


Figure 1.1: Modular programming of the in-house code developed

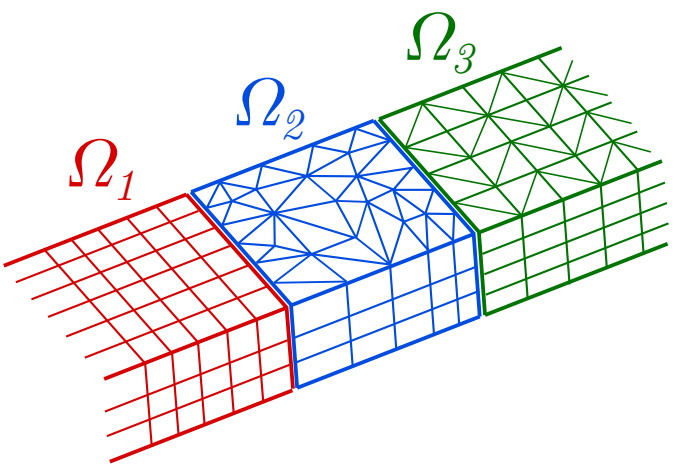


Figure 1.2: Example of different kinds of DDM

that optimal convergence of the iterative solver can be obtained with the so-called Magnetic-to-Electric map on the interface. However, this map is global and it is computationally prohibitive to be found for a general problem, so a new local approximation is proposed in [72]. The second family of DDM developed by Prof. Jin-Fa Lee's group can be understood, indeed, as a local approximation of this map. Moreover, higher-order basis functions have been considered to approximate ancillary variables involved in the definition of TC, [76, 77]. Finally, in [67] a non-overlapping ORAS preconditioner is used with an stochastic-based optimization of the parameters associated to the TC, and an integral representation for absorbing boundary condition (ABC) is included in [78].

One of the most active groups regarding DDM is leaded by Prof. Jin-Fa Lee. Transmission conditions here are classified as first order transmission conditions (FOTC), second order transmission conditions (SOTC)-transversal electric (TE), SOTC-transversal magnetic (TM), and a combination of these last two provides SOTC-FULL, and non-conformality is managed through the definition of some cement variables with physical meaning inspired by mortar element methods, [79–82]. In all the references, the formulation is non-conformal although matching meshes are used up to the definition of a corner-edge penalty term needed to enforce the divergence-free condition, [83, 84]. The convergence of the iterative solver is predicted by a Fourier analysis, [85], also used on [61], and periodicity is taken into account to achieve good performance for this kind of structures, although matching interfaces are used and FOTC (which takes into account only propagating TE and TM modes), as in [86–88]. Details about a parallel implementation of this framework are included in [89]. Massive parallel simulations are also included in [90], although no cement variables are used to implement FOTC. SOTC are first detailed in [91], where Fourier analysis is expanded to this kind of transmission conditions which take into account evanescent TE and TM modes. From this dissertation, and using Gauss-Seidel preconditioning, as in [92], interesting results are detailed in [93–95]. Non-matching interfaces are introduced in [83] through the corner-edge penalty commented above. Using this framework, different variations and results are included in [96] (where mixed-SOTC are proposed to accelerate convergence in some problems using Fourier analysis), [84] (where a different testing to obtain the weak formulation is proposed and block Jacobi is used as preconditioner) and [97]. Recently, in [98] an embedded DDM is

presented to accelerate a number of specific problems, and a technique called reverse operation self-consistent evaluation is used in [99, 100] to improve the numerical integration over non-matching interfaces. Moreover, integral equations with DDM are introduced in [101–104] being an active field of research in the last years; and in [105–107] DDM is used with direct solvers in the reduced surface problem to accelerate conventional direct solvers. Finally, collaboration between these last two families has produced [77] and [58], where a unified framework is developed.

Last family is introduced by Prof. Jian-Ming Jin, developing a machinery which can be summarized in [108]. Dual-primal FETI is expanded to electromagnetism in [109], and a parallel implementation is detailed in [110]. The equivalent to FOTC are included in these references, although an extension to the equivalent SOTC-TE is included in [59], where also an unified framework for conformal and non-conformal DDM is introduced instead of treating all the interfaces as non-conformal, which can lead to some inefficiencies. Comparison with cement element methods from the second family, [83], is also included in [59, 111], obtaining similar results. perfectly matched layers (PML) are introduced in [112], together with an acceleration of the product matrix-vector. Parallel implementation of FETI-DPEM is detailed in [113] with a low-level optimization. An improvement in the convergence of this method is achieved in 2D in [114, 115], introducing PML and three dimensions in [116]. A tree-cotree splitting is introduced in [117] together with DDM to remove low-frequency breakdown, while a new transmission condition based on the normal component of the electric field is proposed in [118]. Multilevel techniques are applied to this family of DDM in [119], obtaining some acceleration. Finally, in [120] performance of iterative solvers when solving the reduced surface problem is improved when having a number of different right-hand side (RHS).

1.2.3 hp adaptivity

Although *hp* adaptivity is included in this work as a proof of concept, some useful references are included in the following. Different estimators can be found in [4, 121–124]. Regarding h adaptive algorithms, red-green technique used for triangular refinement is detailed in [125, 126], and some considerations for the quality of the mesh generated in extruded meshes (including triangular prisms) are introduced in [127]. Refinement with prisms can be found in [128] and in [129], where two

dimensional adaptivity techniques are introduced in three dimensional problems through the use of triangular prisms. Marking strategies for h refinement are found in [6], and for hp refinement in [130], while an analytical study about these strategies is included in [131]. An extensive review of hp adaptivity strategies can be found in [132,133], and details about a possible implementation are found in [27]. Finally, a successful introduction of hp techniques for two-dimensional electromagnetic problems is detailed in [134–136].

1.3 Objective, contributions and organization

The main objective of this dissertation is to study the viability of scalable (in terms of parallel performance) algorithms combining non-conformal DDM and automatic h and hp adaptivity in three dimensions. To do that, three building blocks have to be introduced:

- Basis functions which support non-uniform p refinement.
- A non-conformal, non-overlapping version of DDM.
- An adaptivity algorithm to refine in suitable areas.

The use of non-conformal DDM gives the possibility of having scalable codes in terms of parallelization; specifically, the adaptivity algorithms might be run in each subdomain independently. However, the discontinuities provided by the non-conformal interfaces between subdomains introduce an error which may affect to the refinement of the mesh. This error has not been assessed with detail in the literature and it is a key factor for the adaptivity algorithm to perform properly. In this context, a quantitative analysis of the error due to non-conformal interfaces (including non-conformalities generated by independent refinements in neighbor subdomains) is discussed in this work.

Contributions in this dissertation may be divided in three big blocks:

- Basis functions: a systematic approach has been proposed to develop a new family of basis functions with good mathematical properties. Results about different assembly strategies are included and discussed, and validation of this family of basis functions for the particularization of triangular prisms is included. Also, hierarchical vector basis functions from [26] for tetrahedra, hexahedra and triangular prisms are introduced to ease the non-uniform p refinement needed for hp adaptivity. The introduction of the hierarchical

basis functions has been possible due to the close collaboration between the two research groups in the last years. Validation of the basis functions introduced through MMS is detailed, whose procedure may help to the development of new FEM codes. Results related to the dispersion error (which is a limiting factor for large scale problems) are also included.

- Non-conformal DDM: a number of finite element shapes (hexahedra, tetrahedra and triangular prisms) are introduced to be used with non-overlapping and non-conformal DDM in a parallel fashion. Some variations for MMS are introduced and the procedure is detailed to help the introduction of DDM in other FEM codes. Furthermore, a detailed study of the effect of non-conformal meshes is included, using a rectangular waveguide as wave propagation problem.

- Adaptivity algorithms: a new development of an adaptivity algorithm particularized for the triangular prism introduced in the first block of contributions is detailed. The semi-structured characteristic of this element obliges to include non-standard considerations in the adaptivity algorithm that are shown and justified. Considerations about the use of h adaptivity together with DDM are detailed and some numerical results with MMS are included. A problem with one singularity which is divided into five subdomains is used as a proof of concept, obtaining interesting conclusions when h adaptivity and non-uniform p refinement is applied.

The dissertation is divided in the following chapters: first, formulation on which all these contributions are based are detailed in Chapter 2. Details about the parallel implementation of the code (and particularly, of DDM) are included in Chapter 3. Contributions from the first block are within the Chapter 4, while in Chapter 5 verification and validation of the DDM stated in Chapter 2 are included, with a detailed study of the effect of the non-matching interface in the whole problem covering all the contributions stated in the second block. In Chapter 6 all the contributions from the third block are detailed, while in Chapter 7 a summary of all the conclusions obtained in this dissertation is included together with some future lines.

CHAPTER 2

FEM FORMULATION

The general formulation of the FEM code to solve the vector wave equations obtained through Maxwell's equations used in this document is included in this chapter. First, the boundary value problem is stated; then, Galerkin method is applied obtaining the so-called here *classic* FEM. An expansion of this formulation is introduced to work with different subdomains using non-conformal, non-overlapping DDM inspired by [96], emphasizing on the discretization of the formulation and the two-step solution procedure adopted.

2.1 Variational formulation for *classic* FEM

Maxwell's equations have been used from 1862 to characterize the electromagnetic phenomena, [137]. If the time dependence established in these equations is assumed to be harmonic, the following boundary value problem (BVP) for solving a general open-region electromagnetic problem in a domain Ω in terms of the electric field (a dual case may be obtained with the magnetic field) can be defined as

$$\boldsymbol{\nabla} \times \frac{1}{\mu_r}(\boldsymbol{\nabla} \times \mathbf{E}) - k_0^2 \varepsilon_r \mathbf{E} = \mathbf{O}, \tag{2.1}$$

where $\mathbf{O}$ stands for the term related to the internal sources of the domain. Indeed,

$$\mathbf{O} = -\gamma \eta_0 \mathbf{J} - \boldsymbol{\nabla} \times \frac{1}{\mu_r}\mathbf{M}, \tag{2.2}$$

$\mathbf{J}$ and $\mathbf{M}$ being the electric and magnetic current respectively. For simplicity, isotropic materials are assumed. If anisotropic materials were present, μ_r and ε_r may be represented as tensors.

The boundary conditions considered to close the domain Ω and make unique the solution obtained are homogeneous Dirichlet (used to implement PEC), Neumann (to introduce perfect magnetic conductor (PMC) materials) and Cauchy (to include through $\mathbf{\Psi}$ either the exterior boundary for open region problems or the excitation related to a waveport) introduced as

$$\begin{aligned} \hat{\mathbf{n}} \times \mathbf{E} &= 0, \quad \text{on } \Gamma_{\mathrm{D}}, \\ \hat{\mathbf{n}} \times {\mu_r}^{-1}(\boldsymbol{\nabla} \times \mathbf{E}) &= 0, \quad \text{on } \Gamma_{\mathrm{N}}, \\ \hat{\mathbf{n}} \times {\mu_r}^{-1}(\boldsymbol{\nabla} \times \mathbf{E}) + jk_0 \hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{E}) &= \mathbf{\Psi}, \quad \text{on } \Gamma_{\mathrm{C}}. \end{aligned} \tag{2.3}$$

Variational formulation for (2.1) is derived through Galerkin method, testing with an appropriate weighting function $\mathbf{W} \in \mathbf{H}_0(\mathrm{curl}, \Omega)$ (i.e., in the same space where the electric field belongs, $\mathbf{E} \in \mathbf{H}_0(\mathrm{curl}, \Omega)$) over the whole domain Ω as in

$$\iiint_{\Omega} \mathbf{W} \cdot \boldsymbol{\nabla} \times \left(\mu_r^{-1} \boldsymbol{\nabla} \times \mathbf{E}\right) d\Omega - k_0^2 \iiint_{\Omega} \mathbf{W} \cdot \varepsilon_r \, \mathbf{E} \, d\Omega = \iiint_{\Omega} \mathbf{W} \cdot \mathbf{O} \, d\Omega, \tag{2.4}$$

where $d\Omega$ is the differential volume.

This weak formulation can be rewritten more conveniently if divergence theorem,

$$\iiint_{\Omega} \boldsymbol{\nabla} \cdot \mathbf{A} \, d\Omega = \oiint_{\Gamma} \mathbf{A} \cdot \hat{\mathbf{n}} \, d\Gamma, \tag{2.5}$$

and following vectorial identities,

$$\boldsymbol{\nabla} \cdot (\mathbf{A} \times \mathbf{B}) = \mathbf{B} \cdot \boldsymbol{\nabla} \times \mathbf{A} - \mathbf{A} \cdot \boldsymbol{\nabla} \times \mathbf{B}, \tag{2.6}$$

$$(\mathbf{A} \times \mathbf{B}) \cdot \hat{\mathbf{n}} = -\mathbf{A} \cdot (\hat{\mathbf{n}} \times \mathbf{B}), \tag{2.7}$$

are applied to (2.4). Then, the double curl term in (2.4) can be transformed into

$$\iiint_{\Omega} \mathbf{W} \cdot \boldsymbol{\nabla} \times \left({\mu_r}^{-1} \boldsymbol{\nabla} \times \mathbf{E}\right) d\Omega = \iiint_{\Omega} (\boldsymbol{\nabla} \times \mathbf{W}) \cdot \left({\mu_r}^{-1} \boldsymbol{\nabla} \times \mathbf{E}\right) d\Omega + \oiint_{\partial\Omega} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left({\mu_r}^{-1} \boldsymbol{\nabla} \times \mathbf{E}\right)\right] dS. \tag{2.8}$$

Further simplifications can be applied to this equation. Indeed, last term of (2.8) can be rearranged if boundary conditions are taken into account, i.e.:

$$\oiint_{\partial\Omega} \mathbf{W}\cdot\left[\hat{\mathbf{n}}\times\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)\right]dS = \iint_{\partial\Omega_{\mathrm{D}}} \mathbf{W}\cdot\left[\hat{\mathbf{n}}\times\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)\right]dS+$$
$$\iint_{\partial\Omega_{\mathrm{N}}} \mathbf{W}\cdot\left[\hat{\mathbf{n}}\times\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)\right]dS+\iint_{\partial\Omega_{\mathrm{C}}} \mathbf{W}\cdot\left[\hat{\mathbf{n}}\times\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)\right]dS, \tag{2.9}$$

and since Neumann boundary conditions are applied,

$$\hat{\mathbf{n}}\times{\mu_r}^{-1}(\boldsymbol{\nabla}\times\mathbf{E}) = 0 \rightarrow \iint_{\partial\Omega_{\mathrm{N}}} \mathbf{W}\cdot\left[\hat{\mathbf{n}}\times\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)\right]dS = 0. \tag{2.10}$$

Regarding Dirichlet boundary conditions $\partial\Omega_{\mathrm{D}}$, due to the rearrangement

$$\mathbf{W}\cdot\left[\hat{\mathbf{n}}\times\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)\right] = -\hat{\mathbf{n}}\times\mathbf{W}\cdot\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right), \tag{2.11}$$

then it is straightforward to obtain that:

$$\iint_{\partial\Omega_{\mathrm{D}}} \mathbf{W}\cdot\left[\hat{\mathbf{n}}\times\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)\right]dS = 0. \tag{2.12}$$

Finally, if Cauchy boundary conditions are defined as

$$\hat{\mathbf{n}}\times{\mu_r}^{-1}(\boldsymbol{\nabla}\times\mathbf{E}) = \boldsymbol{\Psi} - jk_0\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\mathbf{E}, \tag{2.13}$$

the surface integral equation in (2.9) simplifies to

$$\oiint_{\partial\Omega} \mathbf{W}\cdot\left[\hat{\mathbf{n}}\times\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)\right]dS = \iint_{\partial\Omega_{\mathrm{C}}} \mathbf{W}\cdot\boldsymbol{\Psi}dS + jk_0\iint_{\partial\Omega_{\mathrm{C}}}(\hat{\mathbf{n}}\times\mathbf{W})\cdot(\hat{\mathbf{n}}\times\mathbf{E})dS. \tag{2.14}$$

Using (2.8) and (2.14) in (2.4), the final weak variational formulation is

$$\iiint_{\Omega} \boldsymbol{\nabla}\times\mathbf{W}\cdot\left({\mu_r}^{-1}\boldsymbol{\nabla}\times\mathbf{E}\right)d\Omega - k_0^2\iiint_{\Omega}\mathbf{W}\cdot\varepsilon_r\,\mathbf{E}\,d\Omega+$$
$$jk_0\iint_{\partial\Omega_{\mathrm{C}}}(\hat{\mathbf{n}}\times\mathbf{W})\cdot(\hat{\mathbf{n}}\times\mathbf{E})dS = \iiint_{\Omega}\mathbf{W}\cdot\mathbf{O}\,d\Omega - \iint_{\partial\Omega_{\mathrm{C}}}\mathbf{W}\cdot\boldsymbol{\Psi}\,dS. \tag{2.15}$$

This formulation can also be expressed through bilinear and linear forms.

Find $\mathbf{E} \in \mathbf{H}_0(\mathrm{curl}, \Omega)$ such that

$$c_1(\mathbf{W}, \mathbf{E}) - k_0^2\, c_2(\mathbf{W}, \mathbf{E}) + jk_0\, c_3(\mathbf{W}, \mathbf{E}) = l(\mathbf{W}), \quad \forall \mathbf{W} \in \mathbf{H}_0(\mathrm{curl}, \Omega), \quad (2.16)$$

with bilinear forms $c_1(\mathbf{W}, \mathbf{E})$, $c_2(\mathbf{W}, \mathbf{E})$, $c_3(\mathbf{W}, \mathbf{E})$ and linear form $l(\mathbf{W})$ as

$$\begin{aligned} c_1(\mathbf{W}, \mathbf{E}) &= \iiint_{\Omega} (\boldsymbol{\nabla} \times \mathbf{W}) \cdot ({\mu_r}^{-1} \boldsymbol{\nabla} \times \mathbf{E})\, d\Omega, \\ c_2(\mathbf{W}, \mathbf{E}) &= \iiint_{\Omega} \mathbf{W} \cdot \varepsilon_r \mathbf{E}\, d\Omega, \\ c_3(\mathbf{W}, \mathbf{E}) &= \iint_{\partial\Omega_{\mathrm{C}}} (\hat{\mathbf{n}} \times \mathbf{F}) \cdot (\hat{\mathbf{n}} \times \mathbf{E})\, dS, \\ l(\mathbf{W}) &= \iiint_{\Omega} (\mathbf{W} \cdot \mathbf{O})\, d\Omega - \iint_{\partial\Omega_{\mathrm{C}}} (\mathbf{W} \cdot \boldsymbol{\Psi})\, dS. \end{aligned} \quad (2.17)$$

To obtain a numerical solution of (2.15), Ω has to be divided into small finite elements where electric field $\mathbf{E}$ is approximated using different sets of vector basis functions. Thus, a set of linear equations $A\mathbf{x} = \mathbf{b}$ is built, where A is a highly sparse square matrix whose bandwidth strongly depends on the order of basis functions used.

2.2 Variational formulation for DDM

Formulation for DDM is defined in the following. In short, the objective of DDM is to create a surface problem (generally not highly sparse) from the whole FEM original problem. This surface problem can be solved with an iterative solver and then, the surface solution is propagated to the interior of each subdomain.

To use this technique, a smooth domain Ω is divided arbitrarily into a number n_{dom} of smaller and disjoints subdomains Ω_i, i.e.,

$$\Omega = \bigcup_{i=1}^{n_{\mathrm{dom}}} \Omega_i, \quad (2.18)$$

with

$$\Omega_i \cap \Omega_j = \emptyset \quad \forall\, i \neq j;\, i, j \in [1, n_{\mathrm{dom}}]. \quad (2.19)$$

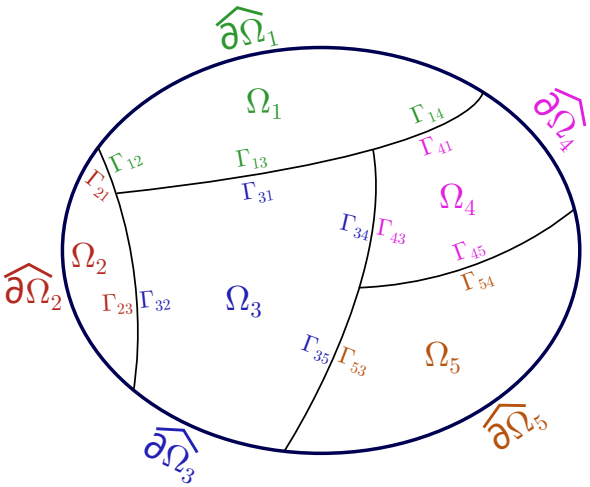


Figure 2.1: A possible non-overlapping division in five subdomains of a smooth domain Ω

For illustration purposes, Figure 2.1 shows a possible division of the original domain Ω into subdomains with $n_{\text{dom}} = 5$, denoting the interface between subdomains i and j belonging to subdomain i as Γ_{ij}. Moreover, exterior boundaries of a subdomain Ω_i can be decomposed into $\partial\Omega_i = \widehat{\partial\Omega}_i \cup \Gamma_{ij}$, with $\widehat{\partial\Omega}_i = \partial\Omega \cap \partial\Omega_i$.

To state the boundary value problem for a non-conformal and non-overlapping DDM, the double curl wave vector equation

$$\boldsymbol{\nabla} \times \frac{1}{\mu_{ri}}(\boldsymbol{\nabla} \times \mathbf{E}_i) - k_0^2 \varepsilon_{ri} \mathbf{E}_i = \mathbf{O}_i \tag{2.20}$$

is defined in terms of the electric field, $\mathbf{O}_i$ being the term related to the internal sources of the domain

$$\mathbf{O}_i = -\gamma\eta_0 \mathbf{J}_i - \boldsymbol{\nabla} \times \frac{1}{\mu_{ri}} \mathbf{M}_i. \tag{2.21}$$

Boundary conditions are imposed over each subdomain Ω_i,

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i = 0, \text{on } \Gamma_{i,\mathrm{D}}, \tag{2.22}$$

$$\hat{\mathbf{n}}_i \times \frac{1}{\mu_{ri}}(\boldsymbol{\nabla} \times \mathbf{E}_i) = 0, \text{on } \Gamma_{i,\mathrm{N}}, \tag{2.23}$$

$$\hat{\mathbf{n}}_i \times \frac{1}{\mu_{ri}}(\boldsymbol{\nabla} \times \mathbf{E}_i) + jk_0\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_i \times \mathbf{E}_i = \boldsymbol{\Psi}_i, \text{on } \Gamma_{i,\mathrm{C}}, \tag{2.24}$$

$$\hat{\mathbf{n}}_i \times \mathbf{E}_i \times \hat{\mathbf{n}}_i = \hat{\mathbf{n}}_j \times \mathbf{E}_j \times \hat{\mathbf{n}}_j, \text{on } \Gamma_{ij}, \tag{2.25}$$

$$\hat{\mathbf{n}}_i \times \frac{1}{\mu_{ri}}(\boldsymbol{\nabla} \times \mathbf{E}_i) = -\hat{\mathbf{n}}_j \times \frac{1}{\mu_{rj}}(\boldsymbol{\nabla} \times \mathbf{E}_j), \text{on } \Gamma_{ij}, \tag{2.26}$$

where well-known Dirichlet, Neumann and Cauchy boundary conditions are introduced in (2.22), (2.23) and (2.24) respectively. Additionally, tangential continuity between subdomains for electric and magnetic fields are enforced in (2.25) and (2.26). This continuity has to be imposed weakly since non-overlapping DDM is used: if it is enforced directly, the subdomain problem may suffer non-negligible internal resonances, [84]. Thus, TC are used to impose this continuity.

The convergence of the surface problem is strongly affected by how these conditions are defined, [58, 84]. In this formulation, SOTC are used,

$$\begin{aligned}\pi_\tau(\mathbf{E}_i) + \pi_\tau^\times(\mu_{ri}^{-1}\boldsymbol{\nabla} \times \mathbf{E}_i) + \beta_i \boldsymbol{\nabla}_\tau \times \boldsymbol{\nabla}_\tau \times \pi_\tau(\mathbf{E}_i) + \\ \gamma_i \nabla_\tau \nabla_\tau \cdot \pi_\tau^\times(\mu_{ri}^{-1}\boldsymbol{\nabla} \times \mathbf{E}_i) = \pi_\tau(\mathbf{E}_j) - \pi_\tau^\times(\mu_{rj}^{-1}\boldsymbol{\nabla} \times \mathbf{E}_j) + \\ \beta_j \boldsymbol{\nabla}_\tau \times \boldsymbol{\nabla}_\tau \times \pi_\tau(\mathbf{E}_j) - \gamma_i \nabla_\tau \nabla_\tau \cdot \pi_\tau^\times(\mu_{ri}^{-1}\boldsymbol{\nabla} \times \mathbf{E}_j)\end{aligned} \tag{2.27}$$

. Particular versions of TC can be obtained from (2.27) with specific values of the constants β_i, γ_i: if $\beta_i, \gamma_i = 0$, FOTC are obtained, while if $\beta_i \neq 0, \gamma_i = 0$, the so-called SOTC for TE modes are defined and, finally, if $\beta_i = 0, \gamma_i \neq 0$ SOTC for TM modes are generated.

To ease the implementation of SOTC, cement variables related to the physics of the problem, [91],

$$\begin{aligned}\mathbf{e}_i =& \pi_\tau(\mathbf{E}_i), \\ \mathbf{j}_i =& \frac{1}{k_0}\pi_\tau^\times\left(\frac{1}{\mu_{ri}}(\boldsymbol{\nabla} \times \mathbf{E}_i)\right), \\ \rho_i =& \frac{1}{k_0}\nabla_\tau \cdot \mathbf{j}_i,\end{aligned} \tag{2.28}$$

are used. In (2.28), $\mathbf{e}_i \in \mathbf{H}(\mathrm{curl}_\tau, \Gamma_{ij})$ is the tangential electric field on the interface, $\mathbf{j}_i \in \mathbf{H}(\mathrm{curl}_\tau, \Gamma_{ij})$ stands for the electric current on the

same interface, and $\rho_i \in H_0^{-1/2}(\Gamma_{ij})$ is a scalar variable with dimensions of charge. First two variables belong to the space $\mathbf{H}(\mathrm{curl}_\tau, \Gamma_{ij})$. It has to be noted that $\mathbf{j}_i$ should reside in $\mathbf{H}(\mathrm{div}_\tau, \Gamma_{ij})$ but a more clustered distribution of eigenvalues (and a faster convergence) is found with $\mathbf{H}(\mathrm{curl}_\tau, \Gamma_{ij})$, [93,118].

Using (2.28) in (2.27), TC can be reformulated through operators,

$$\begin{aligned}(\alpha\mathcal{I}+\beta_i\mathcal{S}_{\mathrm{TE}})(\mathbf{e}_i) + (\mathcal{I}+\gamma_i\mathcal{S}_{\mathrm{TM}})(\mathbf{j}_i) = \\ (\alpha\mathcal{I}+\beta_j\mathcal{S}_{\mathrm{TE}})(\mathbf{e}_j) - (\mathcal{I}+\gamma_j\mathcal{S}_{\mathrm{TM}})(\mathbf{j}_j),\end{aligned} \tag{2.29}$$

where $\mathcal{I}$ is the identity operator, $\mathcal{S}_{\mathrm{TE}} = \boldsymbol{\nabla}_\tau \times \boldsymbol{\nabla}_\tau \times \bullet$ and $\mathcal{S}_{\mathrm{TM}} = \nabla_\tau \nabla_\tau \cdot \bullet$.

Another factor which strongly affects the convergence of the iterative solver in the surface problem is the value of constants α, β_i and γ_i, [91]. One possible choice is, [96],

$$\begin{aligned}\alpha &= -jk_0, \\ \beta_i &= \frac{-1}{\Delta_{\mathrm{TE},i} + jk_0}, \\ \gamma_i &= \frac{1}{k_0^2 - jk_0\Delta_{\mathrm{TM},i}}, \\ \Delta_{\mathrm{TE},i} &= \sqrt{k_{\mathrm{max,TE},i}^2 - k_0^2}, \\ \Delta_{\mathrm{TM},i} &= \sqrt{k_{\mathrm{max,TM},i}^2 - k_0^2}, \\ k_{\mathrm{max,TE},i} &= C_{\mathrm{TE}}\frac{\pi}{h_{\mathrm{min},i}}, \\ k_{\mathrm{max,TM},i} &= C_{\mathrm{TM}}k_{\mathrm{max,TE},i}.\end{aligned} \tag{2.30}$$

The chosen values are functions of the smallest edge in the discretization of the domain, noted as $h_{\mathrm{min},i}$, while C_{TE} and C_{TM} can be adjusted depending on the specific problem and usually they are set to $C_{\mathrm{TE}} = 0.5$ and $C_{\mathrm{TM}} = 2$.

The last step before defining the variational formulation is how to include scalar variables ρ_i. Using Galerkin method, if the definition of ρ_i in (2.28) is tested with a scalar function ϕ_i which belongs to the same space as ρ_i,

$$\iint_{\Gamma_{ij}} \phi_i \rho_j d\Gamma_{ij} - \frac{1}{k_0}\iint_{\Gamma_{ij}} \phi_i \nabla_\tau \cdot \mathbf{j}_i d\Gamma_{ij} = 0, \tag{2.31}$$

integration by parts can be applied, [117], to ease the implementation as

in

$$k_0 \iint_{\Gamma_{ij}} \phi_i \rho_j d\Gamma_{ij} + \iint_{\Gamma_{ij}} \boldsymbol{\nabla}_\tau \phi_i \cdot \mathbf{j}_i d\Gamma_{ij} = 0. \tag{2.32}$$

Finally, if (2.20) is tested with $\mathbf{W}_i$ while (2.29) is tested with $\mathbf{l}_i$, together with (2.32) a variational formulation can be defined analogously as in (2.16).

Find $\mathbf{E}_i \in \mathbf{V}_i, \mathbf{j}_i \in \mathbf{X}_i, \rho_i \in Y_i$ such that

$$\begin{aligned} c_1(\mathbf{W}_i, \mathbf{E}_i) - k_0^2\, c_2(\mathbf{W}_i, \mathbf{E}_i) + \\ jk_0\, c_{\tau,1}(\hat{\mathbf{n}}_i \times \mathbf{W}_i, \hat{\mathbf{n}}_i \times \mathbf{E}_i) = l(\mathbf{W}_i),\ \forall\, \mathbf{W}_i \in \mathbf{V}_i, \end{aligned} \tag{2.33}$$

$$\begin{aligned} \alpha c_{\tau,1}(\mathbf{l}_i, \mathbf{e}_i) + k_0 c_{\tau,1}(\mathbf{l}_i, \mathbf{j}_i) + k_0^2 \gamma_i c_{\tau,1}(\mathbf{l}_i, \boldsymbol{\nabla}_\tau \rho_i) + \\ \beta_i k_0 c_{\tau,1}(\boldsymbol{\nabla}_\tau \times \mathbf{l}_i, \boldsymbol{\nabla}_\tau \times \mathbf{e}_i) = \alpha c_{\tau,1}(\mathbf{l}_i, \mathbf{e}_j) - \\ k_0 c_{\tau,1}(\mathbf{l}_i, \mathbf{j}_j) - k_0^2 \gamma_j c_{\tau,1}(\mathbf{l}_i, \boldsymbol{\nabla}_\tau \rho_j) + \\ \beta_j k_0 c_{\tau,1}(\boldsymbol{\nabla}_\tau \times \mathbf{l}_i, \boldsymbol{\nabla}_\tau \times \mathbf{e}_j),\ \forall\, \mathbf{l}_i \in \mathbf{X}_i, \end{aligned} \tag{2.34}$$

$$c_{\tau,1}(\boldsymbol{\nabla}_\tau \phi_i, \mathbf{j}_i) + k_0 c_{\tau,2}(\phi_i, \rho_i) = 0,\ \forall\, \phi_i \in Y_i, \tag{2.35}$$

where space of functions $\mathbf{V}_i$, $\mathbf{X}_i$ and Y_i are defined as

$$\begin{aligned} \mathbf{W}_i \in \mathbf{V}_i &:= \mathbf{H}_0(\mathrm{curl}; \Omega_i), \\ \mathbf{l}_i \in \mathbf{X}_i &:= \mathbf{H}_0(\mathrm{curl}_\tau; \Gamma_{ij}), \\ \phi_i \in Y_i &:= H_0^{-1/2}(\Gamma_{ij}), \end{aligned} \tag{2.36}$$

and bilinear forms $c_1(\mathbf{F}_i, \mathbf{E}_i)$, $c_2(\mathbf{F}_i, \mathbf{E}_i)$, $c_{\tau,1}(\mathbf{l}_i, \mathbf{e}_j)$, $c_{\tau,2}(\phi_i, \rho_j)$ and linear form $l(\mathbf{F}_i)$ are introduced as

$$\begin{aligned} c_1(\mathbf{W}_i, \mathbf{E}_i) &= \iiint_{\Omega_i} (\boldsymbol{\nabla} \times \mathbf{W}_i) \cdot {\mu_{ri}}^{-1} (\boldsymbol{\nabla} \times \mathbf{E}_i)\, d\Omega_i, \\ c_2(\mathbf{W}_i, \mathbf{E}_i) &= \iiint_{\Omega_i} \mathbf{W}_i \cdot \varepsilon_{ri} \mathbf{E}_i\, d\Omega_i, \\ c_{\tau,1}(\mathbf{l}_i, \mathbf{e}_j) &= \iint_{\Gamma_{ij}} (\mathbf{l}_i \cdot \mathbf{e}_j)\, d\Gamma_{ij}, \\ c_{\tau,2}(\phi_i, \rho_j) &= \iint_{\Gamma_{ij}} (\phi_i \rho_j)\, d\Gamma_{ij}, \\ l(\mathbf{W}_i) &= \iiint_{\Omega_i} (\mathbf{W}_i \cdot \mathbf{O}_i)\, d\Omega_i - \iint_{\Gamma_{i,\mathrm{C}}} (\mathbf{F}_i \cdot \boldsymbol{\Psi}_i)\, d\Gamma_{i,\mathrm{C}} \end{aligned} \tag{2.37}$$

2.2.1 Discretization of the DDM variational formulation

The discretization of the weak formulation defined in (2.33) leads, in the same way as in *classic* FEM, to a system of linear equations $A\mathbf{x} = \mathbf{b}$. However, the division into subdomains leads to a matrix A that can be further decomposed into blocks,

$$\begin{pmatrix} A_1 & C_{12} & \dots & C_{1n} \\ C_{21} & A_2 & \dots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \dots & A_n \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{pmatrix}, \tag{2.38}$$

where A_i is the subdomain i matrix and C_{ij} is the cross matrix between subdomains i and j which stores the interaction between both subdomains. Obviously, $C_{ij} = 0$ if $i \notin \mathcal{N}(j)$, $\mathcal{N}(j)$ being the set of subdomains which are neighbors of the subdomain j.

To get a deeper insight into (2.38), a further division in blocks can be established. For the subdomain matrix A_i,

$$A_i = \begin{pmatrix} A_i^{ii} & A_i^{is} & 0 & 0 \\ A_i^{is} & A_i^{ss} & A_i^{sj} & 0 \\ 0 & A_i^{js} & A_i^{jj} & A_i^{j\rho} \\ 0 & 0 & A_i^{\rho j} & A_i^{\rho\rho} \end{pmatrix}, \tag{2.39}$$

while for the cross matrix C_{ij},

$$C_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & C_{ij}^{js} & C_{ij}^{jj} & C_{ij}^{j\rho} \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.40}$$

Regarding superscripts, $\bullet^i$ is to denote that interior electric field unknowns (i.e. unknowns associated to an entity (vertex, edge, face, volume) out of any interface Γ_{ij}) are involved in the definition of that part of the matrix, $\bullet^s$ is related to surface electric field $\mathbf{e}_i$ unknowns, $\bullet^j$ involves electric current $\mathbf{j}_i$ unknowns and, finally, $\bullet^\rho$ is related to scalar ρ_i variables. The first superscript is referred to the rows of the block matrix, and the second superscript is related to the columns.

Each block matrix is filled by individual terms from (2.33), (2.34) and (2.35). Indeed,

$$
\begin{aligned}
A_i^{ii} =& c_1(\mathbf{W}_i^i, \mathbf{E}_i^i) - k_0^2\, c_2(\mathbf{W}_i^i, \mathbf{E}_i^i) + jk_0\, c_{\tau,1}(\hat{\mathbf{n}}_i \times \mathbf{W}_i^i, \hat{\mathbf{n}}_i \times \mathbf{E}_i^i),\\
A_i^{is} =& c_1(\mathbf{W}_i^i, \mathbf{e}_i) - k_0^2\, c_2(\mathbf{W}_i^i, \mathbf{e}_i) + jk_0\, c_{\tau,1}(\hat{\mathbf{n}}_i \times \mathbf{W}_i^i, \hat{\mathbf{n}}_i \times \mathbf{e}_i),\\
A_i^{si} =& c_1(\mathbf{W}_i^s, \mathbf{E}_i^i) - k_0^2\, c_2(\mathbf{W}_i^s, \mathbf{E}_i^i) + jk_0\, c_{\tau,1}(\hat{\mathbf{n}}_i \times \mathbf{W}_i^s, \hat{\mathbf{n}}_i \times \mathbf{E}_i^i),\\
A_i^{ss} =& c_1(\mathbf{W}_i^s, \mathbf{e}_i) - k_0^2\, c_2(\mathbf{W}_i^s, \mathbf{e}_i) + jk_0\, c_{\tau,1}(\hat{\mathbf{n}}_i \times \mathbf{W}_i^s, \hat{\mathbf{n}}_i \times \mathbf{e}_i),\\
A_i^{sj} =& k_0 c_{\tau,1}(\pi_\tau(\mathbf{W}_i^s), \mathbf{j}_i)\\
A_i^{js} =& \alpha c_{\tau,1}(\mathbf{l}_i, \mathbf{e}_i) + \beta_i k_0 c_{\tau,1}(\boldsymbol{\nabla}_\tau \times \mathbf{l}_i, \boldsymbol{\nabla}_\tau \times \mathbf{e}_i),\\
A_i^{jj} =& k_0 c_{\tau,1}(\mathbf{l}_i, \mathbf{j}_i),\\
A_i^{j\rho} =& k_0^2 \gamma_i c_{\tau,1}(\mathbf{l}_i, \boldsymbol{\nabla}_\tau \rho_i),\\
A_i^{\rho j} =& c_{\tau,1}(\boldsymbol{\nabla}_\tau \phi_i, \mathbf{j}_i),\\
A_i^{\rho\rho} =& k_0 c_{\tau,2}(\phi_i, \rho_i),\\
C_{ij}^{js} =& -\alpha c_{\tau,1}(\mathbf{l}_i, \mathbf{e}_j) - \beta_i k_0 c_{\tau,1}(\boldsymbol{\nabla}_\tau \times \mathbf{l}_i, \boldsymbol{\nabla}_\tau \times \mathbf{e}_j),\\
C_{ij}^{jj} =& k_0 c_{\tau,1}(\mathbf{l}_i, \mathbf{j}_j),\\
C_{ij}^{j\rho} =& k_0^2 \gamma_i c_{\tau,1}(\mathbf{l}_i, \boldsymbol{\nabla}_\tau \rho_j),
\end{aligned}
\tag{2.41}
$$

where the test function $\mathbf{W}_i$ and electric field $\mathbf{E}_i$ unknowns are divided into two sets: interior unknowns ($\bullet^i$) and surface unknowns ($\bullet^s$). Obviously, $\mathbf{E}_i^s = \mathbf{e}_i$ so this last notation is used for clarity purposes.

The block decomposition of A_i and C_{ij} also affects the unknowns $\mathbf{x_i}$ vector, which can be decomposed into

$$
\mathbf{x}_i = \begin{pmatrix} \widetilde{\mathbf{E}_i} \\ \widetilde{\mathbf{e}_i} \\ \widetilde{\mathbf{j}_i} \\ \widetilde{\rho_i} \end{pmatrix}, \tag{2.42}
$$

where $\widetilde{\bullet}$ means unknowns devoted to approximate magnitude $\bullet$; and to the excitations vector, whose block decomposition is

$$
\mathbf{b}_i = \begin{pmatrix} l(\mathbf{W}_i^i) \\ l(\mathbf{W}_i^s) \\ 0 \\ 0 \end{pmatrix}. \tag{2.43}
$$

2.2.2 Two-step solution procedure

More details about the solution of the block matrix defined in (2.38) when DDM is used are given in the following. The idea behind this procedure is to extract the surface unknowns from the original system of equations, solve this reduced and global problem and, then, propagate this unknown to the interior of each subdomain. To perform the extraction, techniques based on the Schur complement can be used. This reduced matrix is not highly sparse, so the performance of direct solvers is worse and iterative solvers are very convenient to use. To assure the convergence of the iterative solver, a preconditioning of the reduced matrix is mandatory. Different preconditioning matrices are available, but one of the most commonly used is the known as Block Jacobi preconditioning matrix,

$$\begin{pmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_n^{-1} \end{pmatrix}. \tag{2.44}$$

Extraction and preconditioning can be performed in the same step, [84], just realizing that the surface problem can be extracted only applying boolean matrices to the resulting matrix. Indeed, if (2.44) is applied to (2.38),

$$\begin{pmatrix} \mathcal{I} & A_1^{-1}C_{12} & \dots & A_1^{-1}C_{1n} \\ A_2^{-1}C_{21} & \mathcal{I} & \dots & A_2^{-1}C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_n^{-1}C_{n1} & A_n^{-1}C_{n2} & \dots & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} A_1^{-1}\mathbf{b}_1 \\ A_2^{-1}\mathbf{b}_2 \\ \vdots \\ A_n^{-1}\mathbf{b}_n \end{pmatrix}, \tag{2.45}$$

is obtained, where $\mathcal{I}$ is the identity matrix.

Then, if the structure of the cross matrix C_{ij} present in (2.40) is recalled, it can be proved that only surface unknowns take part in the preconditioned system of (2.45). For example, if $A_1^{-1}C_{12}$ is expanded

the resulting matrix is

$$\begin{pmatrix} A_1^{ii^{-1}} & A_1^{is^{-1}} & A_1^{ij^{-1}} & A_1^{i\rho^{-1}} \\ A_1^{si^{-1}} & A_1^{ss^{-1}} & A_1^{sj^{-1}} & A_1^{s\rho^{-1}} \\ A_1^{ji^{-1}} & A_1^{js^{-1}} & A_1^{jj^{-1}} & A_1^{j\rho^{-1}} \\ A_1^{\rho i^{-1}} & A_1^{\rho s^{-1}} & A_1^{\rho j^{-1}} & A_1^{\rho\rho^{-1}} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & C_{12}^{js} & C_{12}^{jj} & C_{12}^{j\rho} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1^i \\ \mathbf{x}_1^s \\ \mathbf{x}_1^j \\ \mathbf{x}_1^\rho \end{pmatrix} =$$

$$\begin{pmatrix} 0 & A_1^{ij^{-1}} C_{12}^{js} & A_1^{ij^{-1}} C_{12}^{jj} & A_1^{ij^{-1}} C_{12}^{j\rho} \\ 0 & A_1^{sj^{-1}} C_{12}^{js} & A_1^{sj^{-1}} C_{12}^{jj} & A_1^{sj^{-1}} C_{12}^{j\rho} \\ 0 & A_1^{jj^{-1}} C_{12}^{js} & A_1^{jj^{-1}} C_{12}^{jj} & A_1^{jj^{-1}} C_{12}^{j\rho} \\ 0 & A_1^{\rho j^{-1}} C_{12}^{js} & A_1^{\rho j^{-1}} C_{12}^{jj} & A_1^{\rho j^{-1}} C_{12}^{j\rho} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1^i \\ \mathbf{x}_1^s \\ \mathbf{x}_1^j \\ \mathbf{x}_1^\rho \end{pmatrix} = \tag{2.46}$$

$$\begin{pmatrix} A_1^{ij^{-1}} C_{12}^{js} \mathbf{x}_1^s + A_1^{ij^{-1}} C_{12}^{jj} \mathbf{x}_1^j + A_1^{ij^{-1}} C_{12}^{j\rho} \mathbf{x}_1^\rho \\ A_1^{sj^{-1}} C_{12}^{js} \mathbf{x}_1^s + A_1^{sj^{-1}} C_{12}^{jj} \mathbf{x}_1^j + A_1^{sj^{-1}} C_{12}^{j\rho} \mathbf{x}_1^\rho \\ A_1^{jj^{-1}} C_{12}^{js} \mathbf{x}_1^s + A_1^{jj^{-1}} C_{12}^{jj} \mathbf{x}_1^j + A_1^{jj^{-1}} C_{12}^{j\rho} \mathbf{x}_1^\rho \\ A_1^{\rho j^{-1}} C_{12}^{js} \mathbf{x}_1^s + A_1^{\rho j^{-1}} C_{12}^{jj} \mathbf{x}_1^j + A_1^{\rho j^{-1}} C_{12}^{j\rho} \mathbf{x}_1^\rho \end{pmatrix},$$

where it is seen that $\mathbf{x}_1^i$ does not appear in the final preconditioned vector so, to extract the surface problem, a boolean restriction matrix can be applied to (2.45) selecting the rows and the columns of the unknowns present on the interface problem (i.e., unknowns related to $\mathbf{e}_i$, $\mathbf{j}_i$ and ρ_i).

This surface problem can be solved with an iterative solver or a direct solver depending on the problem and the accuracy that is wanted to be achieved. When the solution of this interface problem is obtained, it can be propagated to each subdomain i with

$$\mathbf{x}_i = A_i^{-1} \left(\mathbf{b}_i - \sum_{j \neq i} \bar{C_{ij}} \bar{\mathbf{x}}_j \right), \tag{2.47}$$

where $\bar{\bullet}$ has been used as an operator to select unknowns related to the interface between subdomains, i.e.,

$$\bar{\mathbf{x}}_i = \begin{pmatrix} \widetilde{\mathbf{e}}_i \\ \widetilde{\mathbf{j}}_i \\ \widetilde{\rho}_i \end{pmatrix}. \tag{2.48}$$

2.3 Conclusions

FEM classic formulation used in Chapter 4 has been detailed, and an expansion to non-overlapping and non-conformal DDM formulation

is introduced by means of cement variables with a physical meaning. Note that the formulation is independent of each finite element shape and, thus, these shapes may be within the same electromagnetic problem. The communication between subdomains is established through cross matrices which are preconditioned to define a global surface problem much smaller than the original classic FEM problem but more populated. This surface problem may be used to construct a two-step solution procedure solving first the surface problem (that may be preconditioned with, e.g., a Block Jacobi preconditioning matrix, to use a matrix-free iterative solver) and then propagating the solution obtained to the interior of each subdomain. The extraction of the surface unknowns from the global problem is also detailed.

CHAPTER 3

THREE-LEVEL PARALLELIZATION

Nowadays, parallelization is a key factor to achieve good performance when solving real electromagnetic problems in FEM. In this work, a three-level parallelization has been implemented:

- Algorithm level: DDM is used for dividing the whole domain into smaller subdomains of controlled size.
- Process level: message passing interface (MPI) is employed to divide the computation between different machines.
- Thread level: open multi-processing (OpenMP) is used to accelerate loops in the same machine.

Regarding DDM, as stated in previous chapters, is an algorithm inherently parallelizable, [69,89,110]. The division in subdomains allows to solve each subdomain with different processes as it will be detailed in this section, and then apply the subsequent two levels to each subdomain.

Well-known libraries used in this thesis such as MUMPS, [138] or MKL, [139] make use of the so-called hybrid parallelism with MPI and OpenMP (last two levels of this three-level parallelization), which is based on the acceleration of the loops in the code through OpenMP directives and the use of MPI to parallelize the same program in different machines. MPI is based on single program multiple data (SPMD), where each process is an instance of the same program, i.e., each process has the same priority and starts and ends at the same time although the stored memory and the code to be run might be different for each process. However, in contrast with MPI paradigm, the memory is shared between

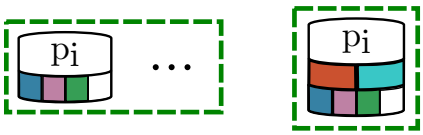


Figure 3.1: Notation for MPI processes in Chapter 3

all the threads of OpenMP (except for private variables) which makes this paradigm more suitable to accelerate loops in the same socket. A typical setting of this kind of parallelization is to use one MPI process per socket with as many threads as the number of cores on that socket.

The unifying thread of this chapter is a flowchart of the execution of the FEM code which is divided into Figs. 3.2, 3.3, 3.4, 3.5, 3.6 and 3.7. Each section includes a thorough explanation of each part of the flowchart. For the sake of clarity, only one frequency and is taken into account in this flowchart but it is straightforward to run the code through different frequencies. Likewise, no details about the adaptivity algorithm are included. In each figure of the flowchart, a process is denoted by a box with one or two rows (see Figure 3.1). In the case of having two rows, the first row is referred to the number of subdomains assigned to one process and the number of threads running at the same time; whereas, in the case of having only one row, it means the number of threads running simultaneously in each step of the flowchart. The distribution of subdomains among processes is further explained in Section 3.1. Finally, when the black dotted line is used to group different processes p_i, the communication in that step uses the MPI global communicator between all the processes; when a thick colored line is present, all the processes included in that area use a (subdomain) communicator between each other (and not the global communicator).

3.1 Assignation of processes to subdomains

The straightforward option to parallelize a DDM problem is to distribute the number of subdomains n_{DOM} among all the n_{MPI} processors. This distribution depends on n_{DOM} and n_{MPI} yielding two cases: *i)* the number of subdomains is higher than the number of processes; and *ii)* the number of processes is higher than the number of subdomains. These two cases are found in Figure 3.2. In the first case, each subdomain is executed *sequentially* from the point of view of MPI, but n_{DOM} subdomains are solved simultaneously. However, in the second case each subdomain

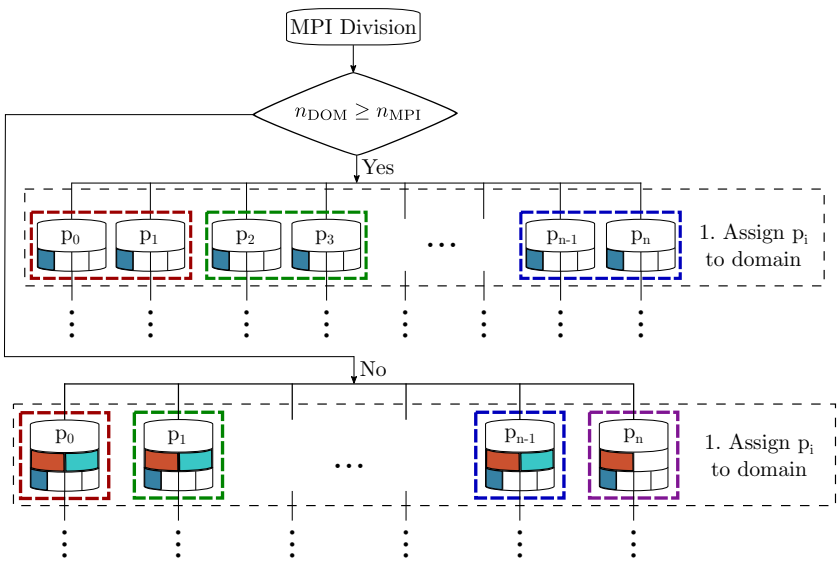


Figure 3.2: Flowchart for the distribution of processes in a DDM problem

is solved *in parallel* with, at least, $\left\lfloor \frac{n_{\text{MPI}}}{n_{\text{DOM}}} \right\rfloor$ processes. This is the most balanced assignation possible in terms of the number of processes solving each subdomain since subdomains are chosen to have similar number of unknowns. In the following figures of the workflow of the simulator, it is assumed that $n_{\text{DOM}} \geq n_{\text{MPI}}$, and that four threads are available to each process.

3.2 Creation of subdomain matrices

Details about the division of the whole geometry problem into processes, keeping only the data necessary for the subdomains assigned to each process, are given in this section.

First, each process reads the input data and the geometry for the whole subdomain. Two possibilities arise to introduce the division in subdomains: *i)* from the input FEM mesh, letting the user decide where to make the divisions; and *ii)* with ParMETIS, [140], which is more flexible and provides optimum subdivisions, although the implementation has to deal with non-planar interfaces. These two possibilities are compatible with each other, so the user can decide where to introduce a first

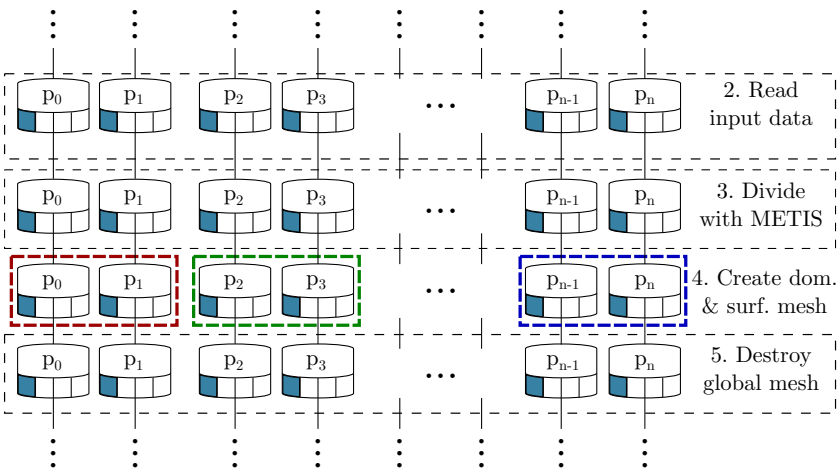


Figure 3.3: Flowchart to build local subdomain matrices

coarse division into subdomains and, then, specify the number of divisions with ParMETIS to apply in each subdomain. This scheme allows an independent mesh in the coarse division and then, applying a finer conformal and optimal division within each subdomain with ParMETIS. Note that, independently of how the division into subdomains is performed, the machinery applied is always non-conformal. Although better performance can be achieved if conformality in the mesh between subdomains is taken into account, when each subdomain is refined iteratively, non-conformality divisions can appear in former conformal divisions.

Thus, in step 2 user-driven division is read from all the processors (although this is not necessary since the distribution in subdomains to each process is known at this point, so in the future this step will be improved) and, in step 3, a second division with ParMETIS is performed to reduce the size of each subdomain. This division is implemented in practice marking the faces which belong to the interface between subdomains and setting a subdomain attribute in the element object as detailed in Section 3.7. When ParMETIS is executed, only a list of elements is obtained. Then, all the faces in the mesh are run and when the same face belongs to different subdomains, it is marked with boundary conditions (BC) which will be created if the interaction between these two subdomains were not registered before. Therefore, a new set of BC not given by the user in the input geometry is created to represent the new

interactions which arise from the conformal division with ParMETIS.

Also, when executing steps 2 and 3 a container with all the elements which belongs to a DDM interface (and classified by interfaces between subdomains) is stored to ease step 4, which consists of building a *local* subdomain mesh object (different for each process) and a global *surface* mesh object (common to all processes which store all the elements belonging to a DDM interface). These mesh objects are containers where all the elements, points and subdomains are located. Since one process can have one or more subdomains, the local subdomain object is indeed an array of mesh objects. This creation of local meshes is due to memory purposes and alleviates the memory stored by each process: indeed, a first approximation to this problem is to build a global mesh object and apply DDM machinery to this global mesh. This last approach is easier for the filling of cross matrices C_{ij} presented in (2.38), but is prohibitive from the point of view of memory when dealing with large scale simulations (which need several computers). For this reason, in this array of local mesh objects only the information related to the subdomain is stored, while in the surface mesh object only information related to the DDM interfaces is present. Finally, in step 5 the global mesh object is deallocated since each process can work independently: to fill the subdomain matrices, the local mesh objects are used, and when interaction with other subdomains is needed (specially to fill cross matrices) the global surface mesh object is used.

3.3 Use of shared points

In the part of the flowchart included in Figure 3.4, some non-conformal DDM structures are created and filled. A classic FEM code needs to number each degree of freedom (DOF) (i.e., to run through all the vertices, edges and faces and assigning the same number for each entity although they belong to different elements) for doing the assembly when filling the matrix to be solved. In step 6, the numbering of DOFs is performed independently for each subdomain and they are classified into four different kinds of DOFs: interior (used to approximate electric field $\mathbf{E}$ out of any interface between subdomains), surface (DOFs which approximate electric field $\mathbf{e}$ in entities which belong to any interface between subdomains), current (used to approximate $\mathbf{j}$ in TC as defined in (2.28)) and charge (scalar DOFs involved in the approximation of ρ as introduced in (2.28)) DOFs. To efficiently perform this classification,

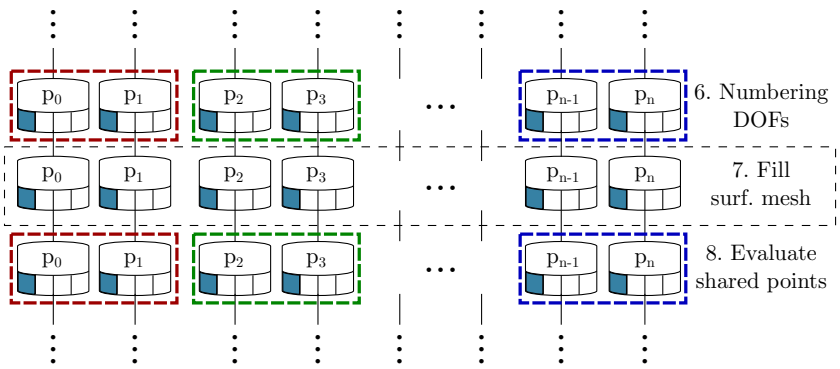


Figure 3.4: Flowchart to implement the interaction between subdomains

vertices (for charge DOFs) and edges are numbered first and, for each face in the element, if it belongs to an interface between subdomains, each DOF related to this face (including not only face but also vertices and edges DOFs) is sorted. When non-planar interfaces (easily generated with ParMETIS division of tetrahedra and triangular prisms) are present, special considerations have to be made. Two different cases have to be taken into account: *i)* an element has two or more faces with the same TC marker, which means that they belong to the same interface between subdomains; and *ii)* different interfaces between subdomains are present in the same element. From the point of view of classification and generation of DOFs, in the first case current DOFs are assembled, i.e., shared edges between faces only have one instance of DOFs (e.g., for order 2, only two DOFs are assigned to one shared edge); while, in the second case, current DOFs are not assembled so, in each face, new DOFs are defined (e.g., for order 2, four DOFs are assigned to one shared edge).

Apart from that, the assignment of some global variables needed for an efficient p refinement (explained in Section 4.6) and the creation of shared points to build cross matrices in (2.38) (used in step 8) are introduced at the same time. Shared points are a procedure to implement non-conformality between subdomains. When a face belongs to an interface between subdomains, a set of shared points that needs to be evaluated by other elements in the neighbor subdomains to implement (2.34) are created. In this way, the numerical integration for one ele-

ment which requires values of the functions from other subdomains can be evaluated for more than one element, which might belong to different subdomains.

Once step 6 is completed by all the processes, numbering and global variables (for p refinement) are communicated through the global MPI communicator to fill the global surface mesh with this numbering and classification of the DOFs. Note that a local numbering is defined for each subdomain and a translation to a global numbering in the surface global mesh is performed in this step.

In step 8, every shared point requested by each subdomain is evaluated. A routine detecting if one shared point is geometrically inside of an element from other subdomain is implemented. When detected, this subdomain is added as neighbor of the requesting subdomain and, apart from the geometry information (subdomain, element, and face on which the point has been detected) of the neighbor element, values of the functions (i.e., $\pi_\tau(\mathbf{N}_i)$ and $\boldsymbol{\nabla}_\tau \times \mathbf{N}_i$ in (2.34)) evaluated by this neighbor element are stored. The search for this element can be introduced as a brute-force search, but more efficient and balanced ways have been implemented: indeed, only elements with a face with the same BC marker as the face on which the shared point has been created are searched; and a certain balance is included in this search. This balance is based on the distribution of the numbers of subdomains: likely, neighbor subdomains are close in the numbering, i.e., for a subdomain i neighbors might be, e.g., $i-2$ and $i+1$. For this reason, elements from subdomains subdomains $i+1$, $i-1$, $i+2$, $i-2$... are searched in this order with modular arithmetic.

Note that step 6 and 8 can be performed in parallel since subdomains are independent of each other thanks to the introduction of the global surface mesh. In these steps, OpenMP acceleration is not used since loops involved here are not time-consuming enough.

3.4 Factorization of subdomain matrices

To perform the factorization of each subdomain matrix, some special considerations are detailed in this section. In Fig 3.5 a summary of the steps to follow is shown.

In step 9, the number of nonzero entries of the matrices to be solved is computed, since every direct solver (MUMPS, PARDISO) supported by the code developed in this dissertation needs a special sparse stor-

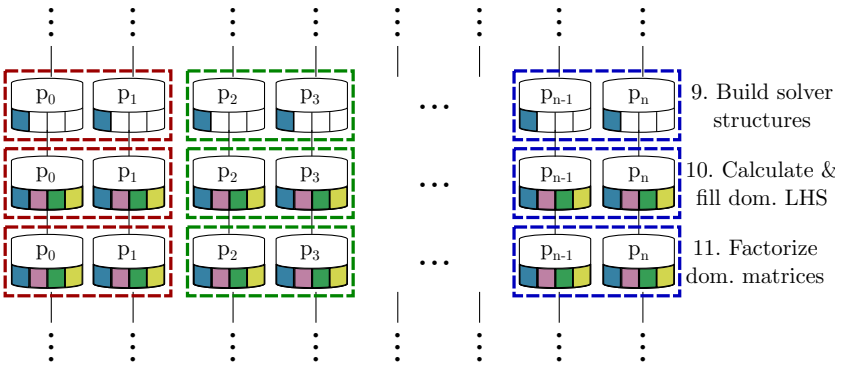


Figure 3.5: Flowchart to factorize subdomain matrices in a DDM problem

age format. Moreover, a machinery to obtain the index and value from the dense matrix is created. Apart from that, the matrix is distributed among the different processes when $n_{\text{DOM}} \geq n_{\text{MPI}}$, so the number of nonzeros is not the same for each process. The distribution of the matrix is implemented by elements, e.g., with 2 MPI processes and 1000 elements, elements from number 1 to number 500 will be assigned to the first process and elements from number 501 to 1000, to the second. Other divisions are possible (e.g., by rows) but no better performance was obtained. The structure for cross matrices C_{ij} is also created, and the same sparse storage format is used due to the high sparsity as shown in (2.40).

The filling of the subdomain and cross matrices is implemented in step 10, where each process fills its portion of the matrix. In step 11, the direct solver performs the factorization of the subdomain matrix with all the distributed portions, storing the factorized object. Regarding cross matrices, same filling of the matrix as for subdomain matrices is performed, but then the matrix is MPI-reduced in all the processes. To alleviate the computational effort of step 12, where the surface matrices are created, a structure storing nonzero columns is created and filled in step 10. To assembly cross matrices, only elements in the global surface mesh are considered, and an efficient loop using the shared points container filled in step 8 is implemented to create the cross matrix C_{ij}.

Note that an OpenMP acceleration has been introduced in step 10 and 11. Regarding step 10, loops for numerical integrations have been

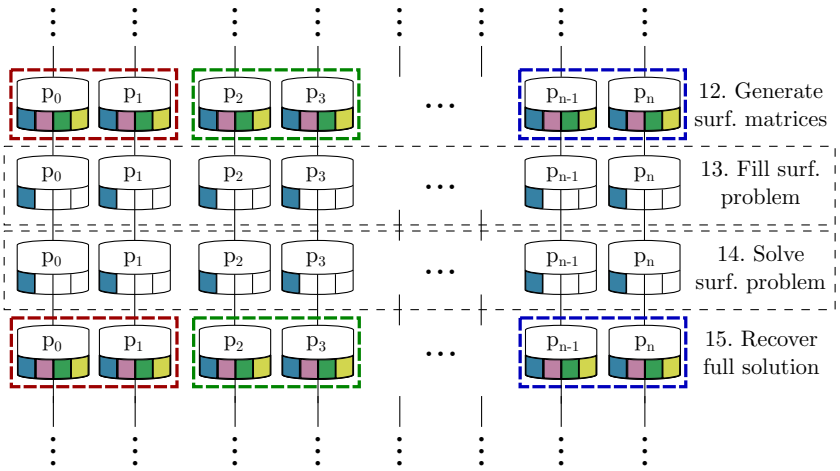


Figure 3.6: Flowchart to solve the surface matrix in a DDM problem

accelerated through multi-threading. With respect to step 11, the parallelization management is performed by the direct solver.

3.5 Solution of the problem

The two-step procedure explained in Section 2.2.2 is developed in steps 12 to 15 as shown in Figure 3.6. First, using factorized objects of each subdomain matrix created in step 11 and cross matrices computed in step 10, surface matrix is generated through (2.45). The explicit generation of this surface matrix is cumbersome and special tricks have to be applied to alleviate the computational effort: sparse RHS may be used when using MUMPS as direct solver; a processing to get a column-ordered cross matrix may be introduced (the transpose of the unordered matrix is transformed through MKL libraries to well-known CSR format, then this matrix is converted to COO format taking rows as columns and columns as rows, since values are ordered through MKL libraries row-wise), and columns may be processed by the direct solver in blocks of 25, which experimentally has been shown as a good tradeoff between time and memory used. However, it is convenient to use a matrix-free iterative solver (as it has been introduced in the code) where this matrix is not explicitly constructed and each iteration is based only in matrix-vector products. The version with the explicit generation of the surface matrix is included for debugging purposes.

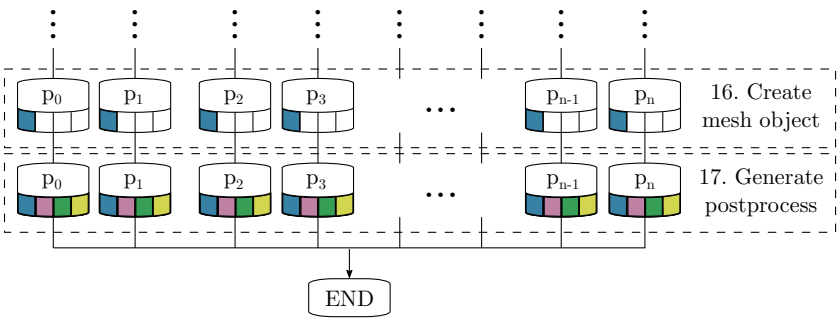


Figure 3.7: Flowchart to implement the postprocessing in a DDM problem

The surface global matrix built in step 12 is not as highly sparse as the subdomain matrices, so an iterative solver is used. In some cases, due to accuracy reasons, this matrix might be solved through a direct solver since the matrix is still sparse. PETSc, [141], is used in steps 13 and 14 as an iterative solver. Since the use of OpenMP threads is not allowed in PETSc, and the distribution of the matrix among processes has to be performed by rows, only MPI processes are employed and the matrix is filled in step 13 through a balanced distribution of rows among all the processes. Finally, the solution obtained through an iterative solver is propagated to the interior of each subdomain with the matrix factorized obtained in step 11 through (2.47).

3.6 Postprocessing and adaptivity

For postprocessing, a global mesh object is created and then, the same module of postprocessing used in classic FEM without DDM is employed, as shown in Figure 3.7. A possible improvement in the future is to implement a postprocessing step separately by subdomains, but the weight in computational time of this step compared to the previous steps is not high enough to justify building a new postprocessing module. Regarding adaptivity, implementation details are included in Chapter 6 due to its distinctive features. In short, when the solution is obtained, an estimator of the error is computed and a refined mesh object is created. With this new mesh object, steps 6 to 15 are repeated until convergence criteria are achieved.

3.7 Data structures

Different data structures are necessary to support the flowchart shown in previous sections. The most important global variables are detailed in the following:

- Array for local subdomain mesh objects, fully independent from other subdomains.
- A global surface mesh object containing only information related to elements belonging to an interface between subdomains.
- Maximum number of DOFs in an element needed for automatic arrays and p adaptivity.
- Maximum number of Gauss points used in numerical integration for any element in the problem.
- Array of size n_{DOM} where, in each position, the *master* process for each subdomain communicator is stored. This array is used for broadcast in intra-domain communications (mainly, to fill the global surface mesh).

On the other hand, the most important Fortran structures (similar to objects in an OOP paradigm) are:

- Mesh, which mainly stores:
  - Array of subdomain objects belonging to this mesh (usually, only one except when a global mesh object is created where an array of subdomains is used).
  - Array of point objects used to represent the elements in the mesh, local numbered.
  - Array of element objects which belong to the mesh.
  - Array of shared points requested in step 8 and filled by other subdomains in step 10.
  - Array of element types which includes the different elements (segment, rectangle, triangle, tetrahedron, hexahedron and triangular prism) supported by the code.

- Domain, which contains an array with the number of elements belonging to that subdomain, the number of total and surface unknowns of the subdomain, an array which stores the neighbor subdomains, the number of non-zeros for the subdomain matrix, arrays which represent sparse format for cross matrices, local RHS, and, finally, the solution of the problem for the subdomain.

- Element type, which is composed of routines related to the geometry of all the shapes supported in the mesh: e.g., the location of points for a given entity (vertices, edges, faces) in the element, the number of points for each entity in the element (very important for hierarchical basis functions of variable order p), and so on.

- Element, which stores the identifiers (references) of the points and boundary conditions which form the geometry of the element, the numbering of DOFs included in the element, the order of each entity, the subdomain in which the element is contained, the identifiers of shared points requested by the element and different arrays to handle non-planar interfaces between subdomains.

- Shared point, which contains the physical point requested by the element on the interface between subdomains, the function values evaluated by the elements in the neighbor subdomains, the Gauss point which generated the shared point and the geometry information (subdomain, element, face) of each element involved in the shared point.

- Solver, which stores the arrays (e.g., sparse storage arrays for subdomain matrices) and parameters involved in the factorization and solution of the system of equations by the direct solver.

3.8 Conclusions

A three-level parallelization of a FEM code is thoroughly explained in this chapter. Details about a balanced distribution of the MPI processes are included, although further improvements can be introduced taking into account the number of unknowns or the refinement in adaptive meshes (i.e., to assign more processes to the subdomains expected to be more refined). The division into subdomains is also explained, with two possible choices (ParMETIS or user-driven through the geometry input data), and the introduction of shared points to communicate the

subdomains is also justified. Finally, a short insight about the variables and structures used in the code to represent the entities presented in this chapter is given in Section 3.7.

CHAPTER 4

BASIS FUNCTIONS

To discretize the FEM formulation introduced in Chapter 2, different kinds of basis functions with a number of distinctive features are found in the literature. As commented in Chapter 1, the choice of basis functions plays a key role in the condition number of the resulting FEM matrix and, thus, in the solving step of the method. In this work, a systematic procedure to obtain basis functions is detailed in Section 4.1, which makes use of the spaces of functions defined in Section 4.2 for tetrahedra, hexahedra and triangular prisms. Then, a set of degrees of freedom as introduced by [21, 22] are defined in Section 4.3 which have to be discretized, i.e., a basis for each space present in the degrees of freedom has to be established and this choice leads to different strategies of assembly explained in Section 4.4. Numerical results comparing these strategies are included. Finally, for this family of basis functions, a comparison with functions from other authors particularized for triangular prisms, and a number of numerical results are detailed in Section 4.5.

However, to ease p refinement, it is convenient to use hierarchical basis functions which allow different approximation orders within the same problem, [26]. In Section 4.6, some details about the introduction of these function in the code are given, and an exhaustive validation of the basis functions using MMS is introduced in Section 6.2, obtaining convergence results of smooth functions. Finally, since the dispersion error may be a limiting factor in some special cases, [142], experiments related to the phase error are included in Section 4.8.

4.1 Procedure

The vector basis functions developed belong to the curl-conforming mixed-order family defined by Nédélec in [21, 22]. Vector instead of scalar basis functions are used to avoid spurious (i.e., non-physical) solutions found when discretizing with FEM the Helmholtz equation (2.1). Curl-conformity means that the tangential continuity between elements is assured, so electromagnetic magnitudes such as electric and magnetic fields (which are tangentially continuous) can be directly approximated by this kind of functions. There exists another family of functions with div-conformity properties, which impose normal continuity between elements, so electric and magnetic inductions can be directly approximated by these functions and electric and magnetic fields should have to be derived from them. Regarding the mixed-order property, this means that the order of polynomial representation of the vector field is not uniform in all directions. The missing terms belong to the null space of the curl operator and they are not involved in the approximation of physical solutions, so they can be removed without loss of accuracy.

If Ciarlet classic definition of finite element is taken into consideration, [143], a geometrical domain, a space of functions, and a set of DOFs have to be defined. A systematic approach is followed here to obtain each finite element: the domain and the space is defined *a priori*, and the definition of DOFs from Nédélec is taken. This definition, used in [21, 22] to prove unisolvency (i.e., the basis functions generated are linearly independent) has to be discretized with basis for each one of the polynomial spaces which appears in that definition.

Then, basis functions are obtained as the dual basis of the degrees of freedom, i.e., discretized DOFs are defined as functionals, $g_i(\mathbf{u})$, and the system of equations

$$g_i(\mathbf{N}_j) = \delta_{ij}, \quad i, j = 1 \dots n_{\mathrm{DOF}} \tag{4.1}$$

is solved $\mathbf{N}_j$ being the basis functions and n_{DOF} the number of DOFs corresponding to each shape and order. In other words, a system of equations of dimension n_{DOF} is solved with n_{DOF} RHS, being the unknowns the coefficients for each basis function. For clarification purposes, a pseudocode of this procedure is shown in Pseudocode 4.1, although a deeper insight particularized for the triangular prism is included in Section 4.3.

With this procedure, each one of the basis functions $\mathbf{N}_i$ is related to one DOF (and, therefore, to a particular region of the element); never-

Pseudocode 4.1 Generation of the values of the coefficients for each basis function

Require: $g_i(\text{entity}, \mathbf{u}, q, \mathbf{q}) \leftarrow$ evaluation of DOF

Input: q_e ▷ Scalar polynomials defined on segment (edge)
Input: **q** ▷ Vector polynomials defined on plane (face)
Output: [COEF] ▷ Coefficient matrix that defines basis functions

1: **procedure** NI_COEF(q_e,**q**,[COEF])
2: Definition of vector monomials $\mathbf{u}_i$ associated with coefficients [COEF], see Pseudocode 4.3
3: Definition of DOF functionals, see Pseudocode 4.4
▷ Coefficients [COEF] obtained by the imposition of (4.1): [A] × [COEF] = [B]
4: $A(i,j) \leftarrow g_i(\mathbf{u}_j)$ ▷ Filling of matrix **A**
5: B $\leftarrow \mathcal{I}_{n_{\text{DOF}}}$ ▷ RHS is identity matrix
6: [COEF] $\leftarrow$ solve([A], [B])
7: **end procedure**

theless, the basis functions here obtained are not interpolatory strictly speaking since interpolatory points are not involved in their definition. The interpolator character and the association of basis functions with certain areas of the element is indirect through the interpolator character of the bases chosen for the discretization.

Although real coordinates can be used with this procedure, to ease the implementation and assembly (see Section 4.4) a reference/master finite element is used. Thus, basis functions are obtained in the master element and transformed into the real element through Jacobian matrix J

$$\mathbf{u} = J^{-1}\mathbf{u}^r \tag{4.2}$$

where $\mathbf{u}^r$ is the vector in the reference element and $\mathbf{u}$ is the mapped vector in the real element (which corresponds to the actual element of the mesh). Further details about this procedure are included in Pseudocode 4.2.

This systematic approach is applied to a number of different shapes as detailed in the following sections: tetrahedron, hexahedron and triangular prism. These spaces of functions have very different characteristics depending on the shape, from the fully unstructured space of the tetra-

Pseudocode 4.2 Computation of the basis functions in the real element from the reference element

Require: real_to_ref_coords(elem, x, y, z)

Input: elem ▷ The element number in the mesh
Input: [COEF] ▷ Coefficient matrix that define basis functions
Input: **r** ▷ Position vector $\mathbf{r} = (x, y, z)$
Output: $\mathbf{N}_i$ ▷ Vector value of each basis function at **r**

1: **procedure** NI_EVAL(elem,[COEF],**r**,$\mathbf{N}_i$)
2: $(x, y, z) \leftarrow \mathbf{r}$
3: $(\xi, \eta, \zeta) \leftarrow$ real_to_ref_coords(elem, x, y, z)

▷ Evaluation of space monomials at (ξ, η, ζ)
4: monomials $\leftarrow 0$ ▷ Initialization of the variable
5: monomials(1, :) $\leftarrow$ Monomials of x component of the space
6: monomials(2, :) $\leftarrow$ Monomials of y component of the space
7: monomials(3, :) $\leftarrow$ Monomials of z component of the space

▷ Evaluation of basis functions in the reference element
8: **for** $i = 1, \ldots, n_{\mathrm{DOF}}$ **do**
9: $N_i(1, i) =$ monomials(1,:) $\times$ COEF(:, i)
10: $N_i(2, i) =$ monomials(2,:) $\times$ COEF(:, i)
11: $N_i(3, i) =$ monomials(3,:) $\times$ COEF(:, i)
12: **end for**

▷ Transformation (4.2) from reference to real element
13: $J \leftarrow$ jacobian_matrix(elem)
14: $\mathbf{N}_i \leftarrow J^{-1}\mathbf{N}_i$

▷ Return of the vector values of the n_{DOF} basis functions
15: $\mathbf{N}_i \leftarrow N_i(1:3, 1:n_{\mathrm{DOF}})$

16: **end procedure**

hedron to the rigid tensor-product based space of the hexahedron, being the triangular prism a hybrid of those spaces. The construction of these spaces affects the performance as it is shown in Section 4.8.

4.2 Spaces of functions

The first step of the systematic approach followed here is to obtain the space of functions for each shape in a way that the order of approximation is not the same in all directions to obtain mixed-order basis functions.

4.2.1 Tetrahedra

Using Nédélec's notation, the space spanned by the basis functions for triangles ($n = 2$) and tetrahedra ($n = 3$) is

$$\boldsymbol{\mathcal{R}_k} = \left\{ \mathbf{u} \in (P_k)^n ; \epsilon^k(\mathbf{u}) = 0 \right\}, \tag{4.3}$$

where $(\mathbf{P}_k)^n$ stands for the n-dimensional full polynomial space of order k and $\epsilon^k(\mathbf{u})$ is a multilinear form which constrains the space, [21], to remove the null space of the curl of order k.

Thus, this operator ϵ^k makes that $\boldsymbol{\mathcal{R}_k}$ lies between the full space of polynomials of order k and order $k-1$, i.e.,

$$(P_{k-1})^n \subset \boldsymbol{\mathcal{R}_k} \subset (P_k)^n, \tag{4.4}$$

so $\boldsymbol{\mathcal{R}_k}$ can be decomposed into the sum of the full space $(P_{k-1})^n$ and a space $\boldsymbol{\mathcal{S}_k}$,

$$\boldsymbol{\mathcal{R}_k} = (P_{k-1})^n \oplus \boldsymbol{\mathcal{S}_k}, \tag{4.5}$$

where $\oplus$ means direct sum between subspaces.

This particular space $\boldsymbol{\mathcal{S}_k}$ is composed of homogeneous polynomials $\mathbf{u}$ of exact order k satisfying the condition $\epsilon^k(\mathbf{u}) = 0$ or, equivalently, $\mathbf{r} \cdot \mathbf{u} = 0$, with $\mathbf{r}$ as the position vector.

A descriptive figure of the properties of a mixed-order space (for a triangle) is shown in Figure 4.1, where the vector $\mathbf{u}$ is decomposed into two components: *i)* $\mathbf{u}_\xi$ along the direction of observation, represented as $\mathbf{r}$, and *ii)* $\mathbf{u}_\eta$ normal to $\mathbf{r}$. It has to be noted that $\mathbf{u}_\xi$ resides in $(P_{k-1})^2$ while $\mathbf{u}_\eta$ resides in $\boldsymbol{\mathcal{S}_k}$ building a mixed-order space $\boldsymbol{\mathcal{R}_k}$ when $\mathbf{u}$ is built from the two normal components.

An explicit expression for the space of functions $\boldsymbol{\mathcal{R}_2}$ for tetrahedra

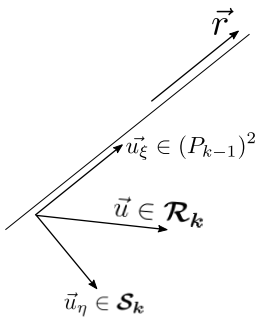


Figure 4.1: Different orders of approximation for Nédélec space

particularized for second order ($k = 2$) is

$$\boldsymbol{\mathcal{R}_2} = \left\{ \begin{array}{l} a_1^{(i)} + a_2^{(i)}\xi + a_3^{(i)}\eta + a_4^{(i)}\zeta + D^{(i)}\eta^2 - F^{(i)}\xi\eta - ... \\ ... - G^{(i)}\xi\zeta + H^{(i)}\zeta^2 + J^{(i)}\eta\zeta \\ \\ b_1^{(i)} + b_2^{(i)}\xi + b_3^{(i)}\eta + b_4^{(i)}\zeta - D^{(i)}\xi\eta - E^{(i)}\eta\zeta + ... \\ ... + F^{(i)}\xi^2 + I^{(i)}\zeta^2 - J^{(i)}\xi\zeta + K^{(i)}\xi\zeta \\ \\ c_1^{(i)} + c_2^{(i)}\xi + c_3^{(i)}\eta + c_4^{(i)}\zeta + E^{(i)}\eta^2 + G^{(i)}\xi^2 - ... \\ ... - H^{(i)}\xi\zeta - I^{(i)}\eta\zeta - K^{(i)}\xi\eta \end{array} \right\}, \quad (4.6)$$

where coefficients $\bullet^{(i)}$ are real coefficients specific for the ith basis function. Obviously, basis functions for tetrahedra $\mathbf{N}_i$ belongs to the same space as $\boldsymbol{\mathcal{R}_2}$ in (4.6), with $i = 1, \ldots, 20$.

4.2.2 Hexahedra

A mixed-order curl-conforming space of functions is built through tensor products (noted as $\otimes$) of one-dimensional polynomials on the reference segment $P_k(I)$ and taking into account the mixed-order property of the space, i.e.,

$$\begin{aligned} \boldsymbol{\mathcal{P}}_{\boldsymbol{k}}^{\text{hexa}} = &(P_{k-1}(I) \otimes P_k(I) \otimes P_k(I)) \\ &\times (P_k(I) \otimes P_{k-1}(I) \otimes P_k(I)) \\ &\times (P_k(I) \otimes P_k(I) \otimes P_{k-1}(I)) \,. \end{aligned} \quad (4.7)$$

An explicit expression for this space particularized for second order

$(k = 2)$ is

$$\boldsymbol{\mathcal{P}}_{\mathbf{2}}^{\text{hexa}} = \left\{ \begin{array}{c} a_1^{(i)} + a_2^{(i)}\xi + a_3^{(i)}\eta + a_4^{(i)}\zeta + a_5^{(i)}\xi\eta + a_6^{(i)}\xi\zeta + ... \\ ... + a_7^{(i)}\eta\zeta + a_8^{(i)}\xi\eta\zeta + a_9^{(i)}\eta^2 + a_{10}^{(i)}\zeta^2 + ... \\ ... + a_{11}^{(i)}\xi\eta^2 + a_{12}^{(i)}\xi\zeta^2 + a_{13}^{(i)}\eta\zeta^2 + a_{14}^{(i)}\eta^2\zeta + ... \\ ... + a_{15}^{(i)}\xi\eta^2\zeta + a_{16}^{(i)}\xi\eta\zeta^2 + a_{17}^{(i)}\eta^2\zeta^2 + a_{18}^{(i)}\xi\eta^2\zeta^2 \\ \\ b_1^{(i)} + b_2^{(i)}\xi + b_3^{(i)}\eta + b_4^{(i)}\zeta + b_5^{(i)}\xi\eta + b_6^{(i)}\xi\zeta + ... \\ ... + b_7^{(i)}\eta\zeta + b_8^{(i)}\xi\eta\zeta + b_9^{(i)}\xi^2 + b_{10}^{(i)}\zeta^2 + b_{11}^{(i)}\eta\xi^2 + ... \\ ... + b_{12}^{(i)z}\eta\zeta^2 + b_{13}^{(i)}\xi\zeta^2 + b_{14}^{(i)}\xi^2\zeta + b_{15}^{(i)}\xi^2\eta\zeta + ... \\ +b_{16}^{(i)}\xi\eta\zeta^2 + b_{17}^{(i)}\xi^2\zeta^2 + b_{18}^{(i)}\xi^2\eta\zeta^2 \\ \\ c_1^{(i)} + c_2^{(i)}\xi + c_3^{(i)}\eta + c_4^{(i)}\zeta + c_5^{(i)}\xi\eta + c_6^{(i)}\xi\zeta + ... \\ ... + c_7^{(i)}\eta\zeta + c_8^{(i)}\xi\eta\zeta + c_9^{(i)}\xi^2 + c_{10}^{(i)}\eta^2 + c_{11}^{(i)}\xi^2\zeta + ... \\ ... + c_{12}^{(i)}\eta^2\zeta + c_{13}^{(i)}\xi\eta^2 + c_{14}^{(i)}\xi^2\eta + c_{15}^{(i)}\xi\eta^2\zeta + ... \\ ... + c_{16}^{(i)}\xi^2\eta\zeta + c_{17}^{(i)}\xi^2\eta^2 + c_{18}^{(i)}\xi^2\eta^2\zeta \\ \\ \end{array} \right\}, \tag{4.8}$$

where the same notation as in (4.6) has been used. The monomials present in (4.8) are straightforward to obtain from the tensor product: e.g., monomials in x-component of $\boldsymbol{\mathcal{P}}_{\mathbf{2}}^{\text{hexa}}$ are obtained from the tensor product $\{1, \xi\} \otimes \{1, \eta, \eta^2\} \otimes \{1, \zeta, \zeta^2\}$. Finally, the dimension of the space is 54 for this kind of element so the space of functions is formed by a set of 54 basis functions $\mathbf{N}_i$, i.e., $\boldsymbol{\mathcal{P}}_{\mathbf{2}}^{\text{hexa}} \equiv \mathbf{N}_i (i = 1, ..., 54)$.

It has to be noted that the resulting space $\boldsymbol{\mathcal{P}}_{\mathbf{2}}^{\text{hexa}}$ defined in (4.8) is not *isotropic* as in the case of tetrahedra, (4.6), where any three of the coordinates can be interchanged obtaining the same space of functions. This characteristic affects the performance of the problems solved with this element, as shown in Section 4.8.

4.2.3 Triangular prism

This element is a hybrid between the former two elements, containing characteristics from the two shapes. Indeed, the space of basis functions for the prism $\boldsymbol{\mathcal{P}}_{\boldsymbol{k}}^{\text{prism}}$ is constructed by taking the tensor product of the mixed-order space for the reference triangle $\boldsymbol{\mathcal{R}}_{\boldsymbol{k}}(T)$ with the space of one-dimensional polynomials on the reference segment $P_k(I)$, i.e.,

$$\boldsymbol{\mathcal{P}}_{\boldsymbol{k}}^{\text{prism}} = (\boldsymbol{\mathcal{R}}_{\boldsymbol{k}}(T) \otimes P_k(I)) \times (\mathbf{P}_k(T) \otimes P_{k-1}(I)), \tag{4.9}$$

where $\mathbf{P}_k(T)$ means the full order space applied to the reference triangle. In (4.9), mixed-order curl conformity is obtained through the properties of $\boldsymbol{\mathcal{R}_k}(T)$ and using one order less $(k-1)$ in the z-component. This space also satisfies commutavitiy of de-Rham diagram, [44].

If (4.9) is particularized for order $k=2$, the space $\boldsymbol{\mathcal{R}_2}(T)$ for triangles is

$$\boldsymbol{\mathcal{R}_2}(T) = \left\{ \begin{array}{c} \alpha_1 + \alpha_2\xi + \alpha_3\eta + \gamma_1\eta^2 - \gamma_2\xi\eta \\ \beta_1 + \beta_2\xi + \beta_3\eta - \gamma_1\xi\eta + \gamma_2\xi^2 \end{array} \right\}, \tag{4.10}$$

while for $\boldsymbol{\mathcal{P}_2}^{\text{prism}}$ is

$$\boldsymbol{\mathcal{P}_2}^{\text{prism}} = \left\{ \begin{array}{c} a_1^{(i)} + a_2^{(i)}\xi + a_3^{(i)}\eta + a_4^{(i)}\zeta + a_5^{(i)}\xi\zeta + ... \\ ... + a_6^{(i)}\eta\zeta + a_7^{(i)}\zeta^2 + a_8^{(i)}\xi\zeta^2 + a_9^{(i)}\eta\zeta^2 + ... \\ ... + C^{(i)}\eta^2 + D^{(i)}\xi\eta + E^{(i)}\eta^2\zeta + F^{(i)}\xi\eta\zeta + ... \\ ... + G^{(i)}\eta^2\zeta^2 + H^{(i)}\xi\eta\zeta^2 \\ \\ b_1^{(i)} + b_2^{(i)}\xi + b_3^{(i)}\eta + b_4^{(i)}\zeta + b_5^{(i)}\xi\zeta + ... \\ ... + b_6^{(i)}\eta\zeta + b_7^{(i)}\zeta^2 + b_8^{(i)}\xi\zeta^2 + b_9^{(i)}\eta\zeta^2 - ... \\ ... - C^{(i)}\xi\eta - D^{(i)}\xi^2 - E^{(i)}\xi\eta\zeta - F^{(i)}\xi^2\zeta ... \\ ... - G^{(i)}\xi\eta\zeta^2 - H^{(i)}\xi^2\zeta^2 \\ \\ c_1^{(i)} + c_2^{(i)}\xi + c_3^{(i)}\eta + c_4^{(i)}\xi^2 + c_5^{(i)}\eta^2 + c_6^{(i)}\xi\eta + ... \\ ... + c_7^{(i)}\zeta + c_8^{(i)}\xi\zeta + c_9^{(i)}\eta\zeta + c_{10}^{(i)}\xi^2\zeta + ... \\ ... + c_{11}^{(i)}\eta^2\zeta + c_{12}^{(i)}\xi\eta\zeta \end{array} \right\}. \tag{4.11}$$

Same notation as in (4.6), (4.8) has been used, and the dimension of this space is 36, i.e., $\boldsymbol{\mathcal{P}_2}^{\text{prism}} \equiv \mathbf{N}_i (i = 1, ..., 36)$. Since this space of functions has been constructed by tensor product, the same property regarding the isotropy of the space (as shown in hexahedra) is obtained here. This affects the performance of the basis functions as shown thoroughly in Section 4.7.

4.3 Degrees of freedom

With the definition of the spaces of functions, DOFs $g_i(\mathbf{u})$ are introduced as functionals to obtain basis functions as dual basis through the imposition of (4.1). In the following, these DOFs are particularized for second order $(k=2)$.

First, DOFs related to edges are defined as

$$g(\mathbf{u}) = \int_e (\mathbf{u} \cdot \hat{\boldsymbol{\tau}}) q \, de, \forall q \in P_1(e), \tag{4.12}$$

where e means *edge*, $\hat{\boldsymbol{\tau}}$ is the unit vector tangent to the considered edge, and $P_1(e)$ is the scalar space of first order polynomials in the corresponding edge local coordinate. There are 2 DOFs per edge.

Then, DOFs associated with faces have to be defined. For second order basis functions, two different DOFs can be present depending on the shape. For triangular faces, present in tetrahedra and triangular prisms, DOF are defined through the expression

$$g(\mathbf{u}) = \iint_{f_t} (\mathbf{u} \times \hat{\mathbf{n}}) \cdot \mathbf{q} \, df_t, \forall \mathbf{q} \in \mathbf{P}_0(f_t), \tag{4.13}$$

where f_t stands for the triangular face considered, $\hat{\mathbf{n}}$ is the outward unit normal vector to f_t, and $\mathbf{P}_0(f_t)$ is the two-dimensional vector space of zero degree polynomials in the two local coordinates of f_t. In each triangular face, 2 DOFs are found.

DOFs associated with quadrilateral faces, present in hexahedra and triangular prisms, are introduced as

$$g(\mathbf{u}) = \iint_{f_q} (\hat{\mathbf{n}} \times \mathbf{u}) \cdot \mathbf{q} \, df_q \quad \forall \mathbf{q} = (q_1, q_2); q_1 \in \mathcal{Q}_{0,1}; q_2 \in \mathcal{Q}_{1,0}, \tag{4.14}$$

where f_q is the quadrilateral face under consideration, $\hat{\mathbf{n}}$ is, again, the outward unit normal vector to f_q and $\mathcal{Q}_{l,m}$ is the two-dimensional scalar vector space of polynomials (i.e., a vector space composed of scalar polynomials) in the corresponding local variables (x_1, x_2) which constitute a basis on the quadrilateral face so that so that the maximum degree is l in x_1 and m in x_2. With this choice of $\mathbf{q}$, the approximation is of one order less in the direction of the component considered. There are 4 DOFs per quadrilateral face.

Finally, for triangular prisms and hexahedra, DOFs associated to the volume (i.e., the interior of the element) have to be introduced. For second order basis functions interior DOFs are not present in tetrahedra and only one kind of interior DOFs is present in triangular prisms, defined as

$$g(\mathbf{u}) = \iiint_V \mathbf{u} \cdot \mathbf{q} \, dV \quad \forall \mathbf{q} \in \mathbf{P}_0(f_t), \tag{4.15}$$

where a vector polynomial $\mathbf{q}$ is defined equivalently as in (4.13). For triangular prisms, there are 2 DOFs of this kind. In the case of hexahedra, DOFs associated with the volume are defined as

$$g(\mathbf{u}) = \iiint_V \mathbf{u} \cdot \mathbf{q}\, dV \quad \forall \mathbf{q} = (q_1, q_2, q_3), \tag{4.16}$$

where $q_1 \in \mathcal{Q}_{1,0,0}, q_2 \in \mathcal{Q}_{0,1,0}$ and $q_3 \in \mathcal{Q}_{0,0,1}$; and $\mathcal{Q}_{l,m,n}$ is the three-dimensional space of scalar vector polynomials in the variables (x_1, x_2, x_3) of the reference hexahedron so that the maximum degree is l in x_1, m in x_2 and n in x_3.

All these DOFs are associated with a basis function as explained in Section 4.1. Thus, basis functions are intrinsically associated with edges, faces and interior volume of the element. Due to the curl-conformity characteristic of the basis functions, in the definition of DOFs related to boundaries (edges and faces), only tangential components are present. Therefore, on a certain part of the boundary, only basis functions associated with that particular entity will give nonzero trace in the tangential sense. As a consequence, basis functions associated with the volume of the element are not involved in the assembly (due to its null trace on the boundaries of the element), and, since all the shapes (tetrahedra, triangular prism and hexahedra) use the same definition of DOFs in edges and faces, meshes with different shapes are possible without non-conformal DDM.

4.3.1 Discretization

When the definition of each DOF is implemented to obtain the basis functions, a basis for each one of the polynomial spaces which appears in these definitions has to be chosen. This process is called here *discretization*, and leads to reference elements as shown in Figure 4.2, 4.3 (where volume DOF are not included for clarity purposes) and 4.4. In all these figures, the numeration of vertices is plotted within a circle, the chosen numeration for DOFs (and basis functions) is included and the length of all the edges parallel to any axis is 1.

Regarding DOFs associated with edges, a basis for $P_1(e)$ has to be chosen. For example, consider the edge from vertex 1 to vertex 2 in any reference element. For that edge, the local coordinate e is ξ, so a possible basis for $P_1(e)$ could be $q_1 = 1$ and $q_2 = \xi$. Another possible basis, with improved properties such as better conditioning of the resulting basis and easier implementation of the assembly (since DOFs can be associated with nodes placed in particular locations of the edge), is to

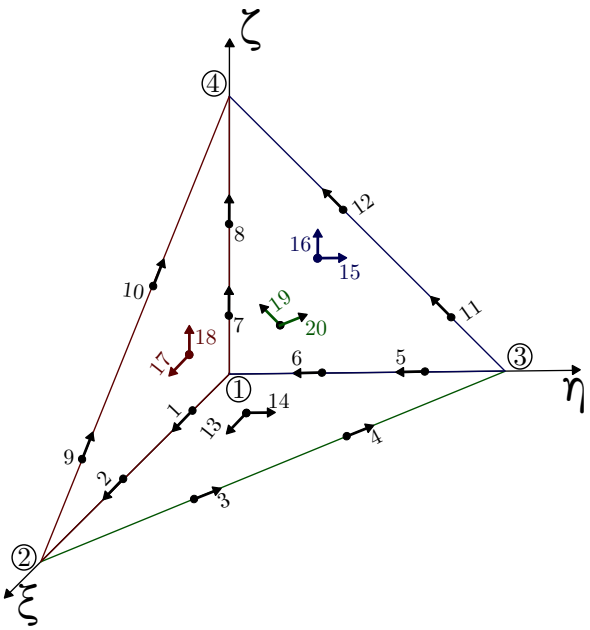


Figure 4.2: Reference tetrahedron of second order

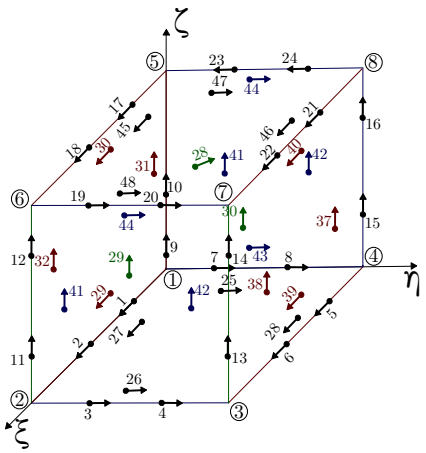


Figure 4.3: Reference hexahedron of second order

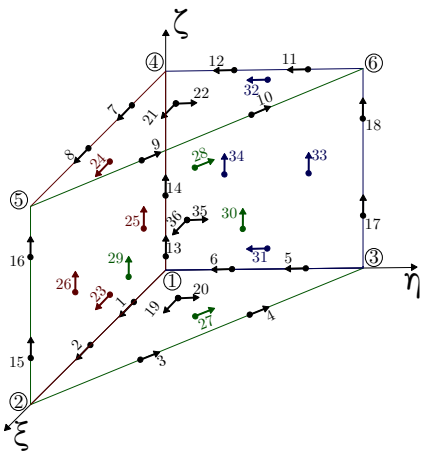


Figure 4.4: Reference triangular prism of second order

choose an interpolatory basis such as Lagrange polynomials, so $q_1 = 1-\xi$ and $q_2 = \xi$. Thus, the first DOF in this edge should be associated with vertex 1 and the second one, with vertex 2, and they are numbered as DOF 1 and 2 in the figures. To make this process more clear, consider the edge from vertex 2 and 3 in Figure 4.4. Here the local coordinate e in the edge is $(1 - \xi + \eta)/2$, so $q_1 = 1 - e = \xi$ and $q_2 = e = \eta$ since $1 - \xi = \eta$ is true along the edge.

In the case of triangular faces, the bases to choose belong to $\mathbf{P}_0(f_t)$, i.e., space of constant vector on the plane formed by the two local coordinates of the face. Thus, $\mathbf{q}$ is reduced to choose two different directions (noted here as $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$) on the plane constituted by these two local coordinates. Consider the case of the face formed by vertex 1, 2, and 3 in Figure 4.2 or, equivalently, any triangular face in the reference prism shown in Figure 4.4, where the local coordinates are ξ and η. In this case, a possible choice is $\hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{\xi}}$ and $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\eta}}$, which corresponds to DOFs 13 and 14 in Figure 4.2, DOFs 19 and 20 for the bottom triangular face in Figure 4.4 and DOFs 21 and 22 for the top triangular face. Different choices and how the performance in the conditioning of the obtained matrices is affected are discussed in Section 4.4.

For quadrilateral faces, the two-dimensional vector $\mathbf{q}$ has to be defined and is not only composed of directions as in the case of triangular faces. For the sake of clarity, let consider the plane $\xi - \eta$, which contains

Table 4.1: Discretization of DOF for the reference tetrahedron

DOF	q	*DOF*	q or $\mathbf{q}$
1	$1-\xi$	11	η
2	ξ	12	ζ
3	ξ	13	$\hat{\boldsymbol{\xi}}$
4	η	14	$\hat{\boldsymbol{\eta}}$
5	η	15	$\hat{\boldsymbol{\eta}}$
6	$1-\eta$	16	$\hat{\boldsymbol{\zeta}}$
7	$1-\zeta$	17	$\hat{\boldsymbol{\xi}}$
8	ζ	18	$\hat{\boldsymbol{\zeta}}$
9	ξ	19	$(-\hat{\boldsymbol{\xi}}-\hat{\boldsymbol{\eta}}+2\hat{\boldsymbol{\zeta}})/2$
10	ζ	20	$(-\hat{\boldsymbol{\xi}}+\hat{\boldsymbol{\eta}})/\sqrt{2}$

the face formed by vertices 1, 2, 4 and 5 in Figure 4.4 and the face formed by vertices 1, 2, 5, and 6 in Figure 4.3. The choice of scalar polynomials q_1 and q_2 has to take into account the mixed-order property of this elements, so a suitable basis for $\mathbf{q}$ is $\mathbf{q}_1 = (1-\zeta)\hat{\boldsymbol{\xi}}$, $\mathbf{q}_2 = \zeta\hat{\boldsymbol{\xi}}$, $\mathbf{q}_3 = (1-\xi)\hat{\boldsymbol{\zeta}}$, $\mathbf{q}_4 = \xi\hat{\boldsymbol{\zeta}}$. If the particular case of the triangular prism is considered, DOFs 23, 24, 25 and 26 can be identified with $\mathbf{q}_1$, $\mathbf{q}_2$,$\mathbf{q}_3$ and $\mathbf{q}_4$ while, for the hexahedron, DOFs 29, 30, 31 and 31 are identified with the same $\mathbf{q}$ defined above. Note the association of these DOFs to different areas and directions in the figures.

Finally, discretization for DOFs associated with the interior volume is defined. In the case of the triangular prism, the space $\mathbf{P}_0(f_t)$ is the same as for triangular faces so the same choice can hold here. Regarding the case of hexahedron, 6 different $\mathbf{q}$ has to be chosen and the same scalar interpolary bases chosen for edges might be an option, so a suitable basis is $\mathbf{q}_1 = (1-\xi)\hat{\boldsymbol{\xi}}$, $\mathbf{q}_2 = \xi\hat{\boldsymbol{\xi}}$, $\mathbf{q}_3 = (1-\eta)\hat{\boldsymbol{\eta}}$, $\mathbf{q}_4 = \eta\hat{\boldsymbol{\eta}}$, $\mathbf{q}_5 = (1-\zeta)\hat{\boldsymbol{\zeta}}$ and $\mathbf{q}_6 = \zeta\hat{\boldsymbol{\zeta}}$.

All these choices for the discretization of DOFs in the reference element are summed up in Tables 4.1, 4.2 and 4.3. Obviously, for different DOFs, the choice of $\mathbf{q}$ can be the same but the g_i is applied to different areas of the finite element so different results are obtained.

With this discretization, now Pseudocode 4.1 can be completed

Table 4.2: Discretization of DOF for the reference hexahedron

DOF	q or $\mathbf{q}$	*DOF*	$\mathbf{q}$
$1, 5, 17, 21$	$1-\xi$	$31, 37$	$(1-\xi)\cdot\hat{\boldsymbol{\zeta}}$
$2, 6, 18, 22$	ξ	$32, 38$	$\xi\cdot\hat{\boldsymbol{\zeta}}$
$3, 7, 19, 23$	1-η	$33, 43$	$(1-\zeta)\cdot\hat{\boldsymbol{\eta}}$
$4, 8, 20, 24$	η	$34, 44$	$\zeta\cdot\hat{\boldsymbol{\eta}}$
$9, 11, 13, 15$	$1-\zeta$	$35, 41$	$(1-\eta)\cdot\hat{\boldsymbol{\zeta}}$
$10, 12, 14, 16$	ζ	$36, 42$	$\eta\cdot\hat{\boldsymbol{\zeta}}$
$25, 47$	$(1-\xi)\cdot\hat{\boldsymbol{\eta}}$	49	$(1-\xi)\cdot\hat{\boldsymbol{\xi}}$
$26, 48$	$\xi\cdot\hat{\boldsymbol{\eta}}$	50	$\xi\cdot\hat{\boldsymbol{\xi}}$
$27, 45$	$(1-\eta)\cdot\hat{\boldsymbol{\xi}}$	51	$(1-\eta)\cdot\hat{\boldsymbol{\eta}}$
$28, 46$	$\eta\cdot\hat{\boldsymbol{\xi}}$	52	$\eta\cdot\hat{\boldsymbol{\eta}}$
$29, 39$	$(1-\zeta)\cdot\hat{\boldsymbol{\xi}}$	53	$\zeta\cdot\hat{\boldsymbol{\zeta}}$
$30, 40$	$\zeta\cdot\hat{\boldsymbol{\xi}}$	54	$(1-\zeta)\cdot\hat{\boldsymbol{\zeta}}$

Table 4.3: Discretization of DOF for the reference triangular prism

DOF	q or $\mathbf{q}$	*DOF*	$\mathbf{q}$
$1, 7$	$1-\xi$	24	$\zeta\cdot\hat{\boldsymbol{\xi}}$
$2, 8$	ξ	25	$(1-\xi)\cdot\hat{\boldsymbol{\zeta}}$
$3, 9$	ξ	26	$\xi\cdot\hat{\boldsymbol{\zeta}}$
$4, 10$	η	27	$(1-\zeta)\cdot(-\hat{\boldsymbol{\xi}}+\hat{\boldsymbol{\eta}})/\sqrt{2}$
$5, 11$	η	28	$\zeta\cdot(-\hat{\boldsymbol{\xi}}+\hat{\boldsymbol{\eta}})/\sqrt{2}$
$6, 12$	$1-\eta$	29	$\xi\cdot\hat{\boldsymbol{\zeta}}$
$13, 15, 17$	$1-\zeta$	30	$\eta\cdot\hat{\boldsymbol{\zeta}}$
$14, 16, 18$	ζ	31	$(1-\zeta)\cdot\hat{\boldsymbol{\eta}}$
$19, 21, 35$	$\hat{\boldsymbol{\xi}}$	32	$\zeta\cdot\hat{\boldsymbol{\eta}}$
$20, 22, 36$	$\hat{\boldsymbol{\eta}}$	33	$\eta\cdot\hat{\boldsymbol{\zeta}}$
23	$(1-\zeta)\cdot\hat{\boldsymbol{\xi}}$	34	$(1-\eta)\cdot\hat{\boldsymbol{\zeta}}$

with Pseudocode 4.3 for the definition of vector monomials and Pseudocode 4.4 for the definition of DOFs as functionals. Since the triangular prism comprises every DOF, these pseudocodes are particularized for that case.

Pseudocode 4.3 Definition of the monomials, in the case of triangular prisms, used in Pseudocode 4.1

Output: $\mathbf{u}$ ▷ Monomials related to the space of the triangular prism $\boldsymbol{\mathcal{P}}_{\mathbf{2}}^{\text{prism}}$

1: **procedure** MONOMIAL_DEFINITION($\mathbf{u}$)
▷ Vector monomials $\mathbf{u}_i$ associated with matrix [COEF]
▷ COEF(:,i) $=[a_1, \ldots, a_9, b_1, \ldots, b_9, C, \ldots, H, c_1, \ldots, c_{12}]$
2: $\mathbf{u}_1 = (1, 0, 0)$ ▷ Vector monomial associated to a_1
⋮
3: $\mathbf{u}_{10} = (0, 1, 0)$ ▷ Vector monomial linked to b_1
⋮
4: $\mathbf{u}_{19} = (\eta^2, -\xi\eta, 0)$ ▷ Vector monomial associated with C
⋮
5: $\mathbf{u}_{36} = (0, 0, \xi\eta\zeta)$ ▷ Vector monomial linked to c_{12}
6: **end procedure**

4.4 Assembly

In a FEM code, the procedure of assembly is based on making equal the values of the matching pairs of DOFs between neighbor elements. This easy procedure is possible here since each basis function is obtained as dual basis of the DOFs functionals which are associated with a certain area of the finite element. Two kinds of DOFs participate in the assembly: edges and faces.

For edges, since DOFs can be considered as vectors with location and direction, the difference in the senses has to be managed through the definition of a global *sign* function. For faces, two cases have to be distinguished: triangular faces (an example of assembly between two elements is shown in Figure 4.5) and quadrilateral faces (another example is illustrated in Figure 4.6). In both figures, numeration of vertices are within circles, while the superscript $\bullet^{(i)}$ in the DOFs denotes the number of the element.

Pseudocode 4.4 Definition of DOFs as functionals in the triangular prism to obtain basis functions

Require: $g_e(\text{edge}, \mathbf{u}, q) \leftarrow$ evaluation of (4.12)
Require: $g_{f_t}(\text{face}, \mathbf{u}, \mathbf{q}) \leftarrow$ evaluation of (4.13)
Require: $g_{f_q}(\text{face}, \mathbf{u}, \mathbf{q}) \leftarrow$ evaluation of (4.14)
Require: $g_v(\mathbf{u}, \mathbf{q}) \leftarrow$ evaluation of (4.15)

Input: $\mathbf{u}$ ▷ Vector monomial related to the basis function
Input: q_e ▷ Scalar polynomials defined on segment (edge)
Input: $\mathbf{q}$ ▷ Vector polynomials defined on plane (face)
Output: $g_i(\mathbf{u})$ ▷ Degrees of freedom defined as functionals

1: **procedure** DOF_DEFINITION($\mathbf{u}$,q_e,$\mathbf{q}$, $g_i(\mathbf{u})$)
2: $g_1(\mathbf{u}) \leftarrow g_e(1, \mathbf{u}, q_1)$ ▷ First DOF of first edge
3: $g_2(\mathbf{u}) \leftarrow g_e(1, \mathbf{u}, q_2)$ ▷ Second DOF of first edge
⋮
4: $g_{18}(\mathbf{u}) \leftarrow g_e(9, \mathbf{u}, q_{18})$ ▷ Second DOF of ninth edge
5: $g_{19}(\mathbf{u}) \leftarrow g_{f_t}(\text{bottom\_face}, \mathbf{u}, \mathbf{q}_{19})$ ▷ First DOF of bottom triangular face
⋮
6: $g_{22}(\mathbf{u}) \leftarrow g_{f_t}(\text{upper\_face}, \mathbf{u}, \mathbf{q}_{22})$ ▷ Second DOF of upper triangular face
7: $g_{23}(\mathbf{u}) \leftarrow g_{f_q}(\text{face } \xi - \zeta, \mathbf{u}, \mathbf{q}_{23})$ ▷ First DOF of face $\xi - \zeta$ (vertices #1-#2-#5-#4)
⋮
8: $g_{35}(\mathbf{u}) \leftarrow g_v(\mathbf{u}, \mathbf{q}_{35})$ ▷ First DOF of volume
9: $g_{36}(\mathbf{u}) \leftarrow g_v(\mathbf{u}, \mathbf{q}_{36})$ ▷ Second DOF of volume
10: **end procedure**

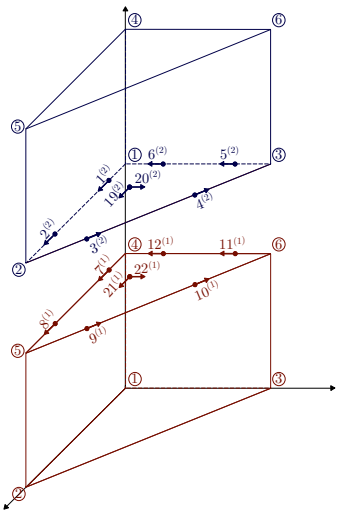


Figure 4.5: Assembly between two prisms matching through one triangular face

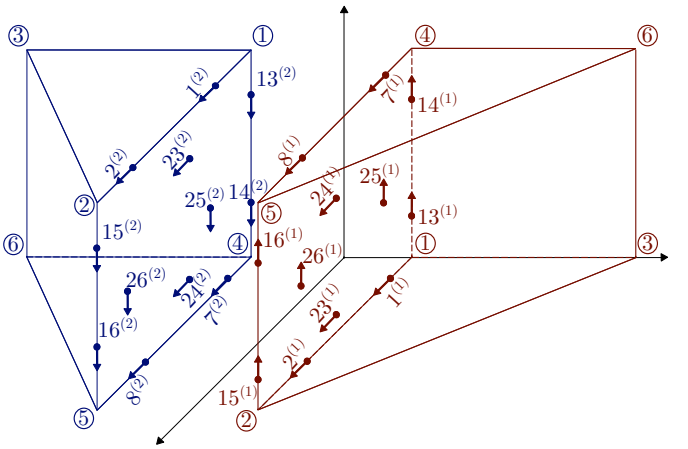


Figure 4.6: Assembly between two prisms matching through one quadrilateral face

To assemble the different basis functions, in Figure 4.6 the equalities $g_{23}^{(1)} = g_{24}^{(2)}$, $g_{24}^{(1)} = g_{23}^{(2)}$, $g_{25}^{(1)} = -g_{25}^{(2)}$ and $g_{26}^{(1)} = -g_{26}^{(2)}$ have to be imposed while, for the Figure 4.5, $g_{19}^{(1)} = g_{21}^{(2)}$ and $g_{20}^{(1)} = g_{22}^{(2)}$. Regarding edges, the difference in the senses can be managed through the imposition of the same global *sign* criterion commented before.

As shown in Section 4.3, different choices of $\mathbf{q}$ to discretize the DOFs are available in this family of basis functions, but they must be taken into account when assembling the elements. For clarification purposes, let us choose an arbitrary $\mathbf{q}$ in the reference element and obtain basis functions for two different neighbor elements of the mesh through Jacobian transformation included in (4.2). In that case, the traces (in the tangential sense) of basis functions of one element and its neighbor on a shared face of the mesh will not be equal in general and curl-conformity of the finite element approximation will be lost. Two strategies to deal with this issue are proposed and compared in the following.

4.4.1 *vq* assembly

The so-called in this dissertation *vq* strategy consists of choosing unique sets of $\mathbf{q}$ on each face of the mesh and transforming them to the reference element using the Jacobian matrix for each one of the two finite elements sharing that face. An example of this strategy is shown in Figure 4.7a. Vector $\mathbf{q}^r$ will denote $\mathbf{q}$ converted to the reference element. In general, different $\mathbf{q}^r$ will be obtained for one element and its neighbor. Thus, the coefficients of the basis functions need to be computed for each element, although the choice of $\mathbf{q}$ on one face of the element only affects the coefficients associated with the basis functions of that face. In the code, this strategy is implemented in parallel by a finite element initialization routine that loops over every element in the mesh and calculates the coefficients of the basis functions for each element. More details can be found in Pseudocode 4.5, where Pseudocode 4.1 is updated with the two different assembly strategies. Thus, curl-conformity is preserved while arbitrary directions for pairs of $\mathbf{q}$ on each (triangular or quadrilateral) face can be chosen. Specifically, orthogonal directions for $\mathbf{q}$ (denoted as $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$, see Figure 4.7) may be defined with a positive impact in the conditioning of the resulting finite element matrices in some cases as it is shown in Section 4.4.3. This strategy has been proven better for tetrahedra than for triangular prisms and hexahedra. In the last two shapes, the choice of $\mathbf{q}$ in (4.14) seems to provide a better performance when they are parallel to the edges involved in the tensor

product than when they are orthogonal. This is due to the construction of the space based on the tensor product for these shapes, [21].

4.4.2 *vc* assembly

The other strategy, denoted here as *vc*, is based on choosing pairs of $\mathbf{q}$ on each face of the mesh in such a way that each $\mathbf{q}$ vector is parallel (directions $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$) to the edges of the face. It can be proven that with this choice of $\mathbf{q}$, transformation to the real element yields to $\mathbf{q}^r$ with identical components parallel to the edges of the face, i.e., same edge trace is obtained on both neighbor elements sharing a common face of the mesh. This feature allows to construct a limited set of pairs of $\mathbf{q}^r$ in the reference element such that the coefficients are obtained once in the reference element, being valid for all the elements of the mesh. In some sense, this strategy resembles the case of working with basis functions expressed in affine coordinates, and can be implemented by a simple initialization routine, as shown in Pseudocode 4.5. This initialization routine computes a super-set of basis functions coefficients corresponding to the different pairs of directions that can be chosen on a face in the general case. For example, directions parallel to the edges 1-2 (from vertex 1 to vertex 2), edges 2-3 or edges 3-1 for the triangular face (see Fig 4.7b for more details), and parallel to edges 1-2, edges 2-3, edges 3-4, or edges 4-1 for the quadrilateral face might be selected. Later, the routine that performs the filling of the FEM matrices through a numerical integration selects a pair of edge directions for each face of the mesh based on some global criterion (typically, choosing the pair $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ the closest to the orthogonal case as possible). Thus, curl-conformity is preserved without computing different coefficients for each element of the mesh. However, the behavior of the *vc* strategy concerning the condition number with severe elongated elements may be not as good as with *vq* strategy previously described as it is shown in Section 4.4.3. This difference between strategies arises because the directions $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ may be far from orthogonal since they are necessarily chosen to be parallel to the edges of the face.

In the case of assembling quadrilateral faces, not only directions ($\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$) as in the case of triangular faces but also directions and scalar polynomials (q) must match between elements. As an example, in Figure 4.6, the DOF associated with $\mathbf{q}_{24}^{(1)}$ has to match to $\mathbf{q}_{23}^{(2)}$ instead of $\mathbf{q}_{24}^{(2)}$ although they have the same direction. This particularity is implemented through a local parametrization of arbitrary directions.

Pseudocode 4.5 Generation of numerical values for the coefficients depending on the assembly strategy chosen (*vc* or *vq*)

Require: $g_i(\mathbf{u}, \mathbf{q}) \leftarrow$ evaluation of (4.12)-(4.14)

Require: $[\text{COEF}]_{\text{ref}}(\mathbf{q}^r, \text{coords}) \leftarrow$ select precalculated [COEF] in reference element

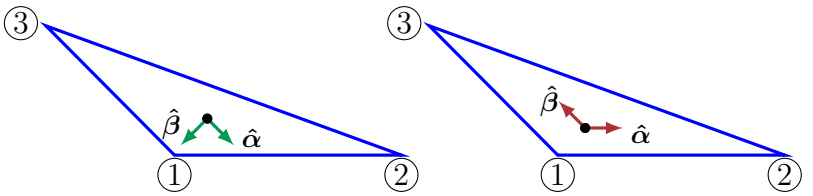
Input: vq ▷ Flag to choose *vq* or *vc* version

Input: coords ▷ Coordinates from real element

Input: global_num ▷ Global numeration of vertices

Output: [COEF] ▷ Coeff. matrix that defines basis functions

1: **procedure** NI_COEF(vq,coords,global_num,[COEF])
▷ In each face $\mathbf{q}$ are defined
2: **for** face $\leq n_{\text{faces}}$ **do**
▷ If *vq* version, $\mathbf{q}$ are chosen in the real element
3: **if** vq is selected **then**
▷ $\mathbf{q}_{f,vq}$ returns two orthogonal vectors on the face
4: $\mathbf{q}_{\text{real}} \leftarrow \mathbf{q}_{f,vq}$(coords,face)
▷ $\mathbf{q}^r$ in ref. element obtained through Jacobian matrix J
5: $\mathbf{q}^r \leftarrow J \times \mathbf{q}_{\text{real}}$
6: **else**
▷ $\mathbf{q}_{f,vc}$ returns two vectors parallel to the edges
7: $\mathbf{q}^r \leftarrow \mathbf{q}_{f,vc}$(global_num)
8: **end if**
9: **end for**
10: **if** vq is selected **then**
▷ Use of (4.1) to obtain [COEF].
11: $A(i,j) \leftarrow g_i(\mathbf{u}_j, \mathbf{q}^r)$ ▷ Filling of matrix $\mathbf{A}$
12: $B \leftarrow \mathcal{I}_{n_{\text{dof}}}$ ▷ RHS is identity matrix
13: [COEF] $\leftarrow$ solve(A, B)
14: **else**
▷ [COEF] are selected from the precalculated family
15: [COEF] $\leftarrow [\text{COEF}]_{\text{ref}}(\mathbf{q}^r, \text{coords})$
16: **end if**
17: **end procedure**



(a) *vq* **(b)** *vc*

Figure 4.7: Illustration of the two assembly-oriented strategies for basis functions design

4.4.3 Numerical experiments

To estimate the best assembly strategy for all the shapes, numerical results of the condition number of the FEM matrix, with different distortions in the shape of the finite element are included and discussed in the following.

The condition number is defined as the ratio of the maximum and minimum eigenvalue of the matrices corresponding to the inner products of the vector basis functions $\boldsymbol{N}_i$ in a finite element —i.e., element mass matrix M—,

$$M_{ij} = c_2(\boldsymbol{N}_i \cdot \boldsymbol{N}_j), \tag{4.17}$$

and of their curls, —i.e., stiffness matrix K—,

$$K_{ij} = c_1(\boldsymbol{N}_i \cdot \boldsymbol{N}_j). \tag{4.18}$$

It has to be noted that numerically null eigenvalues of the stiffness matrix are discarded.

Normalized versions of the basis functions involved are used in order to balance possible differences in conditioning only due to scale factors. Thus, preconditioned matrices M^p and K^p are defined, respectively, as

$$M^p = D^{-1}MD^{-1}, K^p = D^{-1}KD^{-1}, \tag{4.19}$$

where D is a diagonal matrix with $D_{ii} = \sqrt{M_{ii}}$.

The tetrahedron is considered first. In this case, the deformation is introduced by changing the z coordinate of the top vertex of the reference element while holding the same bottom face. Specifically, the coordinates of the four vertices are compactly expressed in Table 4.4. The results obtained are included in Table 4.5 and 4.6. Three cases of the *vc* version

Table 4.4: Coordinates of tetrahedra used in the analysis of the condition number

Tetrahedron deformation	
Vertex	Coordinates
$\boldsymbol{r}_1$	$(0,0,0)$
$\boldsymbol{r}_2$	$(1,0,0)$
$\boldsymbol{r}_3$	$(0,1,0)$
$\boldsymbol{r}_4$	$(0,0,\varepsilon)$

Table 4.5: Condition numbers for tetrahedron deformation (mass matrix)

	Reference	*Tetrahedron deformation*		
	tetrahedron	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 16$
Version	$[M^p]$	$[M^p]$	$[M^p]$	$[M^p]$
vc	38	185	670	2615
vc	66	464	1657	6411
vc	70	142	505	1964
vq	34	154	547	2124

of the basis functions proposed in this paper are considered depending if the directions $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ of vectors $\mathbf{q}$ are chosen parallel to edges 1-2, 2-3 or 3-1 of the triangular faces of the tetrahedron. Only one case of the *vq* version of the basis functions is shown, where directions $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ are chosen orthogonal in all faces. It may be observed how the *vq* version shows a better performance than the *vc* version, as expected. In the particular case of deformation considered here, one of the *vc* versions corresponds to the choice of orthogonal directions for $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ and, hence, the *vc* strategy is very similar to the *vq* assembly.

The triangular prism is considered next. Due to the hybrid character of its space of functions and the fact that contains triangular and quadrilateral faces two deformations have been defined: "triangle deformation" and "rectangle deformation". In the first case, triangular top and bottom faces of the prisms are stretched out whereas quadrilateral faces are kept rectangular. In the second case, the prism under study is generated

Table 4.6: Condition numbers for tetrahedron deformation (stiffness matrix)

	Reference	*Tetrahedron deformation*		
	tetrahedron	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 16$
Version	$[K^p]$	$[K^p]$	$[K^p]$	$[K^p]$
vc	12	69	670	2615
vc	12	108	422	1676
vc	14	66	246	969
vq	11	66	246	965

Table 4.7: Coordinates of the prisms used in the analysis of the condition number for the two cases under study

Triangle deformation		*Rectangle deformation*	
Vertex	Coordinates	Vertex	Coordinates
$\boldsymbol{r}_1$	$(0, 0, 0)$	$\boldsymbol{r}_1$	$(0, 0, 0)$
$\boldsymbol{r}_2$	$(\varepsilon, 0, 0)$	$\boldsymbol{r}_2$	$(1, 0, 0)$
$\boldsymbol{r}_3$	$(0, 1, 0)$	$\boldsymbol{r}_3$	$(0, 1, 0)$
$\boldsymbol{r}_4$	$(0, 0, 1)$	$\boldsymbol{r}_4$	$(2, 2, 1/\kappa)$
$\boldsymbol{r}_5$	$(\varepsilon, 0, 1)$	$\boldsymbol{r}_5$	$(3, 2, 1/\kappa)$
$\boldsymbol{r}_6$	$(0, 1, 1)$	$\boldsymbol{r}_6$	$(2, 3, 1/\kappa)$

by extrusion in an inclined direction non-orthogonal to the triangular base. Thus, rectangular faces of the reference element are transformed into parallelograms more and more stretched out. The position of the vertices in these two cases are summarized in Table 4.7.

Condition numbers are shown in Tables 4.8, 4.9, 4.10, and 4.11. Tables 4.8 and 4.9 show condition numbers obtained with the triangle deformation for the mass and stiffness matrices, respectively. As in the case of the tetrahedron, three cases of the *vc* version are considered for the two triangular faces (with the same directions $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ selected on the upper and bottom faces), while the *vq* version is included with $\hat{\boldsymbol{\alpha}} \perp \hat{\boldsymbol{\beta}}$. Different choices of orthogonal pairs of $\mathbf{q}$ have been studied, providing similar condition numbers.

It is clearly observed that the *vq* strategy always returns better condi-

Table 4.8: Condition numbers for prisms under triangle deformation (mass matrix)

	Reference	*Triangle deformation*		
	prism	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 16$
Version	$[M^p]$	$[M^p]$	$[M^p]$	$[M^p]$
vc	81	1587	18826	276385
vc	81	217	738	2827
vc	71	215	737	2825
vq	72	215	737	2826

Table 4.9: Condition numbers for prisms under triangle deformation (stiffness matrix)

	Reference	*Triangle deformation*		
	prism	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 16$
Version	$[K^p]$	$[K^p]$	$[K^p]$	$[K^p]$
vc	37	210	791	3096
vc	37	199	733	2856
vc	38	197	732	2854
vq	37	197	732	2854

Table 4.10: Condition numbers for prisms under rectangle deformation (mass matrix)

	Reference	*Rectangle deformation*		
	prism	$\kappa = 2$	$\kappa = 4$	$\kappa = 8$
Version	$[M^p]$	$[M^p]$	$[M^p]$	$[M^p]$
vc	72	3107	12270	48926
vq	72	2187	8435	33432

tion numbers compared with each *vc* version. Note that in the particular case of the triangle deformation considered here, there is one *vc* combination that corresponds to orthogonal directions and, hence, it provides

Table 4.11: Condition numbers for prisms under rectangle deformation (stiffness matrix)

	Reference prism	*Rectangle deformation*		
		$\kappa = 2$	$\kappa = 4$	$\kappa = 8$
Version	$[K^p]$	$[K^p]$	$[K^p]$	$[K^p]$
vc	37	2566	10205	40765
vq	37	2066	8171	32599

identical conditioning to those of the *vq* strategy as in the case of tetrahedra. In the case of arbitrarily distorted elements, the difference in the condition number, even between the best *vc* combination and the *vq* strategy, may be remarkable. Thus, it is concluded that the best choice for $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ when defining DOFs in triangular faces,(4.13), is that $\hat{\boldsymbol{\alpha}} \perp \hat{\boldsymbol{\beta}}$.

Tables 4.10 and 4.11, show the results for the rectangle deformation. Due to the particular case of rectangle deformation considered here in which the rectangular faces are parallelograms, only one *vc* case is possible. Again, regarding *vq* strategy orthogonal directions for $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ are selected. Similar comments to those just made about the impact of triangle deformation in the condition number hold here. However, as it is observed in the tables, the effect of orthogonality in the *vq* vectors has a lower impact in the case of rectangle deformation, at least, for the deformation considered.

The different performance observed between triangle and rectangle deformations is a consequence of the tensor product structure of the prismatic space of functions that somewhat hybridizes tetrahedron and hexahedron types of behavior.

In this context, and to draw definite conclusions about the choice of $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ in (4.14), the case of the hexahedron is considered. The deformations used in the analysis are expressed in Table 4.12. The condition numbers obtained are shown in Tables 4.13, 4.14, 4.15, and 4.16. In this case, the *vc* strategy is better than the *vq* strategy showing the influence of the tensor-product nature of the space of functions in the condition number. For prisms, in Tables 4.10 and 4.11 the results obtained with both strategies are similar and, since here the DOFs to take into account

Table 4.12: Coordinates of the hexahedra used in the analysis of the condition number

Two-face deformation		*Four-face deformation*	
Vertex	Coordinates	Vertex	Coordinates
$\boldsymbol{r}_1$	$(0,0,0)$	$\boldsymbol{r}_1$	$(0,0,0)$
$\boldsymbol{r}_2$	$(1,0,0)$	$\boldsymbol{r}_2$	$(1,0,0)$
$\boldsymbol{r}_3$	$(0,1,0)$	$\boldsymbol{r}_3$	$(1,1,0)$
$\boldsymbol{r}_4$	$(0,1,0)$	$\boldsymbol{r}_4$	$(0,1,0)$
$\boldsymbol{r}_5$	$(2,0,1/\kappa_1)$	$\boldsymbol{r}_5$	$(2,2,1/\kappa_2)$
$\boldsymbol{r}_6$	$(3,0,1/\kappa_1)$	$\boldsymbol{r}_6$	$(3,2,1/\kappa_2)$
$\boldsymbol{r}_7$	$(2,1,1/\kappa_1)$	$\boldsymbol{r}_7$	$(3,3,1/\kappa_2)$
$\boldsymbol{r}_8$	$(2,0,1/\kappa_1)$	$\boldsymbol{r}_8$	$(2,3,1/\kappa_2)$

Table 4.13: Condition numbers for hexahedra with two-face deformation (mass matrix)

	Reference	*Rectangle deformation*		
	hexahedron	$\kappa_1 = 2$	$\kappa_1 = 4$	$\kappa_1 = 8$
Version	$[M^p]$	$[M^p]$	$[M^p]$	$[M^p]$
vc	19	912	3552	14112
vq	19	1503	5923	23607

are the same as in the hexahedra case, the only difference is the space of functions, being the tensor product of hexahedra "stricter" than the one used for prisms. Thus, the main conclusion that can be drawn here is that *vq* strategy is discouraged for elements whose space of functions is constructed through a tensor product, whereas for simplices based on (4.3) *vq* strategy shows a better performance.

Table 4.14: Condition numbers for hexahedra with two-face deformation (stiffness matrix)

	Reference	*Rectangle deformation*		
	hexahedron	$\kappa_1 = 2$	$\kappa_1 = 4$	$\kappa_1 = 8$
Version	$[K^p]$	$[K^p]$	$[K^p]$	$[K^p]$
vc	30	2131	8721	35168
vq	30	2155	8738	35182

Table 4.15: Condition numbers for hexahedra with four-face deformation (mass matrix)

	Reference	*Rectangle deformation*		
	hexahedron	$\kappa_2 = 2$	$\kappa_2 = 4$	$\kappa_2 = 8$
Version	$[M^p]$	$[M^p]$	$[M^p]$	$[M^p]$
vc	19	1869	7405	29552
vq	19	2696	10531	41883

Table 4.16: Condition number for hexahedra with four-face deformation (stiffness matrix)

	Reference	*Rectangle deformation*		
	hexahedron	$\kappa_2 = 2$	$\kappa_2 = 4$	$\kappa_2 = 8$
Version	$[K^p]$	$[K^p]$	$[K^p]$	$[K^p]$
vc	30	3689	14616	58318
vq	30	4553	17970	71635

4.5 Comparison with other families of triangular prisms

To measure the robustness of some basis functions, the condition number of the resulting FEM matrices is commonly used. In this section, robustness of the family of basis functions developed here and particularized for triangular prisms is compared against two representative sets of basis functions.

Basis functions provided in [54] and a version inspired in [57] are included. A second order version of the basis functions of [54] and a second mixed-order prism inspired in the spectral element proposed in [57] (using Lagrange interpolatory polynomials instead of polynomials based on Gauss-Legendre-Lobatto integration points) have been introduced in the code. Specifically, the bases coded for the last case are

$$\begin{aligned}
L_m L_l^2 \mathbf{W}_{ij};\ & i,j=1,2,3; j>i; m=i,j; l=4,5,\\
L_i^2 L_l \boldsymbol{\nabla} L_l;\ & i=1,2,3; l=4,5,\\
L_k L_l^2 \mathbf{W}_{ij};\ & i,j,k=1,2,3; j>i; k\neq i,j; l=4,5,\\
L_m L_l L_{l+1} \mathbf{W}_{ij};\ & i,j=1,2,3; j>i; m=i,j; l=4,\\
L_i L_j L_l \boldsymbol{\nabla} L_l;\ & i,j=1,2,3; j>i; l=4,5,\\
L_k L_l L_{l+1} \mathbf{W}_{ij};\ & i,j,k=1,2,3; j>i; k\neq i,j; l=4,
\end{aligned} \tag{4.20}$$

where the set (L_1, L_2, L_3) corresponds to the affine coordinates for the triangle and (L_4, L_5) are the affine coordinates for the segment which connects triangular faces; finally, $\mathbf{W}_{ij}$ stands for the Whitney functions $L_i \nabla L_j - L_j \nabla L_i$. The third subset of bases are face functions associated to the upper and bottom triangular faces. For each face, only two of the three possible bases must be selected. The sixth subset of bases are interior functions and only two basis functions must be chosen.

The three families to be compared (interpolatory bases provided by Graglia, spectral bases and the family obtained with a systematic approach proposed here) are checked to span Nédélec mixed-order space for the prism described in expression (4.11). Second order basis functions of [54] (denoted there as $p=1$) and those of (4.20) are based on an explicit construction using Whitney functions multiplied by appropriate polynomials. In the case of [54], polynomials are constructed based on fully interpolatory properties on a number of points on the whole prism (while basis functions proposed in this dissertation are not interpolatory in that sense). The case of functions described in (4.20) is somewhat

Table 4.17: Condition number of mass matrices for different cases of prisms (triangle deformation)

	Reference	*Triangle deformation*		
	prism	$\kappa = 4$	$\kappa = 8$	$\kappa = 16$
Version	$[M^p]$	$[M^p]$	$[M^p]$	$[M^p]$
vq	72	215	737	2826
Interpolatory,(2-3)+(1-2)	33	173	638	2497
Interpolatory,(3-1)+(1-2)	37	174	639	2498
Spectral(3-1)	301	1020	3967	15871
Spectral(2-3)	171	1021	3967	15871
Spectral(1-2)	171	842	3468	14046

halfway between those two. Basis functions are naturally associated to edges, faces and interior, and interpolatory polynomials used on the explicit construction of the basis are defined in terms of points only on the corresponding entity (edges, faces or volume) but not over the whole prism.

All the matrices taken here into consideration are normalized following (4.19). In the case of functions of [54], the specific normalization factors proposed by their authors are used prior to the pre-scaling using $[D]$. In all cases, the condition number after preconditioning by $[M^p]$ and $[K^p]$ is always improved.

Same kind of deformations specified in Tab. 4.7 are used on this analysis. Condition numbers are shown in Tables 4.17, 4.18, 4.19 and 4.20, only showing the vq version (choosing orthogonal $\mathbf{q}$ vectors) for basis functions proposed in this dissertation. Moreover, different combinations of basis functions of [54] and (4.20) associated to triangular faces have been considered. Basis functions of [54] provide the best conditioning in all cases. Basis functions proposed here are competitive with respect to [54] in terms of condition number, since both sets of basis functions provide the same order of magnitude for the mass and stiffness matrices, whereas basis functions of (4.20) have the worst condition number for both types of deformation, with roughly one order of magnitude higher than with the other sets of basis functions.

Table 4.18: Condition number of stiffness matrices for different cases of prisms (triangle deformation)

	Reference	*Triangle deformation*		
	prism	$\kappa = 4$	$\kappa = 8$	$\kappa = 16$
Version	$[K^p]$	$[K^p]$	$[K^p]$	$[K^p]$
vq	37	197	732	2854
Interpolatory,(2-3)+(1-2)	16	102	394	1555
Interpolatory,(3-1)+(1-2)	19	104	394	1551
Spectral(3-1)	24	108	417	1659
Spectral(2-3)	20	109	418	1659
Spectral(1-2)	20	101	398	1588

Table 4.19: Condition number of mass matrices for different cases of prisms (rectangle deformation)

	Reference	*Rectangle deformation*		
	prism	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$
Version	$[M^p]$	$[M^p]$	$[M^p]$	$[M^p]$
vq	72	2187	8435	33432
Interpolatory	37	1484	5889	23509
Spectral	301	5967	23559	93928

Table 4.20: Condition number of stiffness matrices for different cases of prisms (rectangle deformation)

	Reference	*Rectangle deformation*		
	prism	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$
Version	$[K^p]$	$[K^p]$	$[K^p]$	$[K^p]$
vq	37	2066	8171	32599
Interpolatory	19	1067	4279	17131
Spectral	24	1209	4226	16923

4.6 Hierarchical basis functions

To ease the introduction of p refinement, hierarchical vector basis functions from [3] have been included in the code. In short, a new family of high order shape functions with two important features are introduced: *i)* fully compatibility between all the shapes and *ii)* non-uniform p refinement within the same element. Among the four classical energy spaces (H^1, $H(\text{curl})$, $H(\text{div})$ and L^2) available in [3], only curl-conforming ($H(\text{curl})$) and interpolatory (H^1) are included in the code. With these two families all the spaces of functions introduced in Chapter 2 can be represented. Mathematical properties are thoroughly explained by the authors in [3]; in this section, only some details about the introduction of functions with non-uniform p order in the code are explained.

First, since the order of the basis functions is no longer fixed to $p = 2$, some considerations about the size of the structures of the code have to be taken into account in order to have an efficient use of memory. Three options for memory management are available in Fortran: *i)* dynamic arrays (also called allocatable) which are allocated and deallocated with instructions `ALLOCATE` and `DEALLOCATE` called by the user; *ii)* static arrays whose size is known at compile time; and *iii)* automatic arrays whose size is known at run time and allowed from Fortran 2003. The difference between *ii)* and *iii)* is that the dimension of the array in the former is fixed by some parameter (or even a constant number) set at compile time while in the latter the dimension is determined by some variable which can change in runtime. Regarding performance, both are

similar since they are allocated in the stack. With respect to the use of dynamic arrays, a little degradation in time and memory performance can be found if the size of the arrays is not large enough; however, there is no limit on the stack space for them since they are not allocated in this space.

In the code, variables with size depending on the number of DOFs per entity (edge, face or volume) and on the number of Gauss points for numerical integration are mandatory in the filling of the FEM matrix. The straightforward option is to establish some parameter called `MAX_NUM_DOF` and `MAX_NUM_GAUSS` at compile time setting a maximum order to be run. However, this leads to a waste of resources if lower orders are used since the increase in the order means an exponential growth in these parameters. Furthermore, the management of dynamic arrays is tedious from the point of view of the implementation and not a good solution either since the size of these arrays are in the order of 500 for polynomials of order 5. Therefore, automatic arrays are used and the value of these two parameters are determined in the numbering of DOFs (step 6 in the flowchart of Chapter 3).

Secondly, since in DDM formulation DOFs related to some edge or face have to be extracted, a variable order has to be taken into account. When only second order basis functions are used, with a lookup table is enough; however, in the case of non-uniform order within the element, orders in each entity have to be checked and then, the correct number of DOFs and their local numeration have to be selected within the element.

Moreover, a different assembly strategy has to be coded. For this family of hierarchical vector basis functions, the concept of *orientation* is thoroughly described in [3]: in short, this orientation allows the assembly between elements just defining the relative position of the element with respect to its neighbor. In the implementation introduced in the code, this is performed when numbering DOF. In this loop through all elements, the first element which appears sets the reference for each edge and face, and the subsequent elements sets the orientation accordingly to the reference comparing the relative position of the vertices.

Finally, special considerations about the imposition of non-homogeneous Dirichlet BC for this family are detailed in the following. For each edge, the following system of n_e equations (with n_e equal to the number of DOFs in the edge),

$$A\mathbf{x}_{\mathrm{D}} = \mathbf{b}, \tag{4.21}$$

has to be solved, where $\mathbf{x}_\mathrm{D}$ are the values imposed for each DOF with Dirichlet, and left-hand side (LHS) matrix A is

$$A_{ij} = \int_e (\mathbf{N}_i \cdot \hat{\boldsymbol{\tau}})(\mathbf{N}_j \cdot \hat{\boldsymbol{\tau}}) de \quad \forall i, j = 1, \ldots, n_e, \tag{4.22}$$

while RHS $\mathbf{b}$ is

$$b_i = \int_e (\mathbf{N}_i \cdot \hat{\boldsymbol{\tau}})(\mathbf{d} \cdot \hat{\boldsymbol{\tau}}) de \quad \forall i, j = 1, \ldots, n_e, \tag{4.23}$$

where, in both equations, e stands for the edge affected by the Dirichlet BC, $\mathbf{N}_i$ is one of the basis functions related to e, $\hat{\boldsymbol{\tau}}$ is the unit vector tangential to e and $\mathbf{d}$ is the value of the non-homogeneous Dirichlet BC.

Regarding faces, another system of equations taking into account $\mathbf{x}_\mathrm{D}$ for edges is formulated, i.e.,

$$A\mathbf{x}_\mathrm{D} = \mathbf{b}, \tag{4.24}$$

where the dimension of matrix A is $n_f + h_f$, with n_f as the number of DOFs on the face f, and h_f as the number of scalar H^1 DOFs in the same face f. Each position of the LHS matrix A is defined as

$$A_{ij} = \iint_f ((\boldsymbol{\nabla} \times \mathbf{N}_i) \cdot \hat{\mathbf{n}})((\boldsymbol{\nabla} \times \mathbf{N}_i) \cdot \hat{\mathbf{n}}) df \quad \forall i, j = 1, \ldots, n_f, \tag{4.25}$$

$$A_{i,n_f+j} = \iint_f (\pi_\tau(\mathbf{N}_i) \cdot \boldsymbol{\nabla}_\tau \phi_j) df \quad \forall i = 1, \ldots, n_f; j = 1, \ldots, h_f, \tag{4.26}$$

and

$$A_{n_f+i,j} = \iint_f (\boldsymbol{\nabla}_\tau \phi_j \cdot \pi_\tau(\mathbf{N}_i)) df \quad \forall i = 1, \ldots, h_f; j = 1, \ldots, n_f. \tag{4.27}$$

And RHS vector $\mathbf{b}$ is formulated with

$$b_i = \iint_f ((\boldsymbol{\nabla} \times \mathbf{N}_i) \cdot \hat{\mathbf{n}}) \cdot (\mathbf{d}_\nabla \cdot \hat{\mathbf{n}}) df \quad \forall i = 1, \ldots, n_f, \tag{4.28}$$

and

$$b_{n_f+i} = \iint_f (\boldsymbol{\nabla}_\tau \phi_i) \cdot \pi_\tau(\mathbf{d}) df \quad \forall i = 1, \ldots, h_f \tag{4.29}$$

where $\mathbf{d}_\nabla$ is the curl of the Dirichlet value to be imposed without the contributions of the edges belonging to that face, i.e.,

$$\mathbf{d}_\nabla = \boldsymbol{\nabla} \times \mathbf{d} - \sum_k x_k \cdot (\boldsymbol{\nabla} \times \mathbf{N}_k) \quad \forall k = 1, \ldots, n_e. \tag{4.30}$$

More details about this procedure are found in [27].

4.7 Verification with MMS

Apart from showing the robustness of the developed basis functions, they have to be verified to show the correctness of the approximation of the field. This verification is usually implemented by comparing with well-known analytic results but here MMS, [13], is used. This method is based on manufacturing an analytical solution to some differential equation by solving the problem backwards. If an equation of the type $\mathcal{D}(\mathbf{u}) = \mathbf{f}$ —where $\mathcal{D}$ is the differential operator, $\mathbf{u}$ is the vector solution and $\mathbf{f}$ is the source term— has to be solved, $\mathbf{u}$ is manufactured and then the operator is applied to find $\mathbf{f}$.

Formulation defined in Section 2.1 is tested next, so an analytic electric field $\mathbf{E}$ (denoted here as $\mathbf{E}_{\text{MMS}}$) is obtained and $\mathbf{O}$ in (2.1) and $\mathbf{\Psi}$ in (2.3) are computed. Then, $\mathbf{O}$ and $\mathbf{\Psi}$ are plugged into the FEM code through (2.16) and a numerical approximated solution $\mathbf{E}_{\text{FEM}}$ is obtained. Finally, the energy error between $\mathbf{E}_{\text{MMS}}$ and $\mathbf{E}_{\text{FEM}}$ is computed in the field itself and its curl separately, i.e.,

$$\varsigma = \frac{\| c_2 \left((\mathbf{E}_{\text{FEM}} - \mathbf{E}_{\text{MMS}}) , (\mathbf{E}_{\text{FEM}} - \mathbf{E}_{\text{MMS}})^* \right) \|_2}{\| c_2 \left(\mathbf{E}_{\text{MMS}}, \mathbf{E}^*_{\text{MMS}} \right) \|_2}, \tag{4.31}$$

$$\varsigma_{\text{curl}} = \frac{\| c_1 \left((\mathbf{E}_{\text{FEM}} - \mathbf{E}_{\text{MMS}}) , (\mathbf{E}_{\text{FEM}} - \mathbf{E}_{\text{MMS}})^* \right) \|_2}{\| c_1 \left(\mathbf{E}_{\text{MMS}}, \mathbf{E}^*_{\text{MMS}} \right) \|_2}. \tag{4.32}$$

In this section, three kinds of results are included: *i)* simple test of the machinery of the code; *ii)* test of the full variational formulation included in (2.1); and *iii)* test with smooth functions. The first two tests use monomials belonging to the space of functions and the error obtained is expected to be numerically zero, while in the last test smooth functions such as exponential functions are included to get convergence results. Specifically, for the last case, the error is expected to converge asymptotically to zero when the number of elements is increased. As predicted by the theory (for example, [1, 144]) the behavior of the error in the asymptotic regime is error $= C\, h^p$ where h stands for the diameter of the element.

For the sake of brevity, the whole test bench is not included for all the shapes implemented in the code. Results *i),ii)* and *iii)* are included for triangular prism developed in this dissertation, results *i)* and *ii)* are applied to the tetrahedra with hierarchical vector basis functions and, finally, result *ii)* is tested with hexahedra and prisms implemented with the same hierarchical vector basis functions from Section 4.6.

First test consist of introducing only Dirichlet BC, which is the simplest equation that tests the machinery of the code with the least number of terms possible: in (2.16), $c_1(\mathbf{W},\mathbf{E})$ and $c_3(\mathbf{W},\mathbf{E})$ are set to 0 and only Dirichlet BC are imposed through

$$\hat{\mathbf{n}} \times \mathbf{E} = \hat{\mathbf{n}} \times \mathbf{E}_{\text{MMS}}, \quad \text{on } \Gamma_{\text{D}}. \tag{4.33}$$

Second and third tests are implemented through the Cauchy BC and machinery and formulation are checked at the same time. In this case, the formulation is the same as for a real problem except for $\mathbf{\Psi}$ in (2.3) which now is called $\mathbf{\Psi}_{\text{C}}$ and defined as

$$\mathbf{\Psi}_{\text{C}} = \hat{\mathbf{n}} \times \frac{1}{\mu_r}(\boldsymbol{\nabla} \times \mathbf{E}_{\text{MMS}}) + jk_0 \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{E}_{\text{MMS}}, \quad \text{on } \Gamma_{\text{C}}. \tag{4.34}$$

In the case of second test, $\mathbf{E}_{\text{MMS}}$ is every monomial inside of the space of basis functions while for the third test a smooth function such a complex exponential function is used as manufactured solution, i.e.,

$$\mathbf{E}_{\text{MMS}} = \mathbf{E}_{\text{pol}} e^{-jk_0(\hat{\mathbf{k}}_p \cdot \boldsymbol{r})}, \tag{4.35}$$

where $\mathbf{E}_{\text{pol}}$ is the polarization vector, $\hat{\mathbf{k}}_p$, the unit propagation vector and $\mathbf{r}$ the position vector.

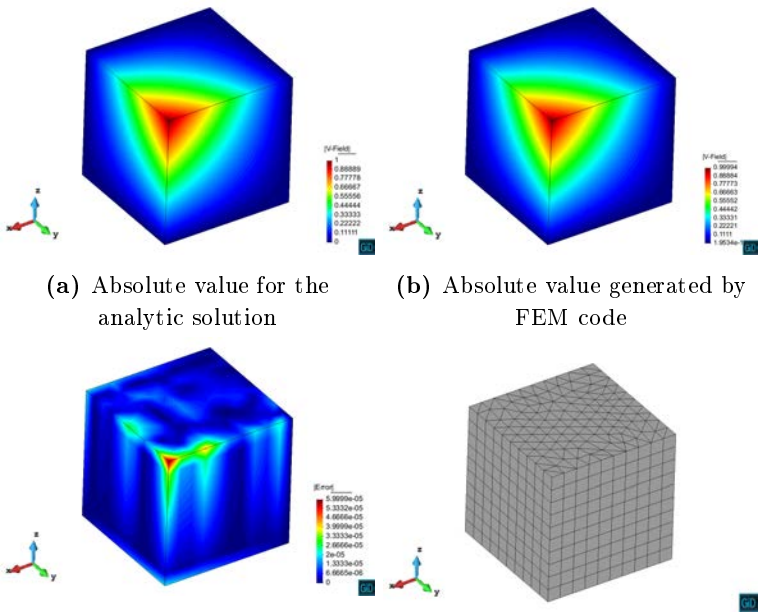
4.7.1 Systematic triangular prism

Basis functions obtained with the systematic approach introduced in Section 4.1 are tested in this section. Regarding first and second tests, they are applied to the 36 monomials belonging to the space of basis functions obtaining similar results for all of them; for brevity purposes, only one monomial (related to coefficient $G^{(i)}$) is chosen here to illustrate results, i.e.,

$$\mathbf{E}_{\text{MMS}} = \left\{ \begin{array}{c} y^2 z^2 \\ -xyz^2 \\ 0 \end{array} \right\}. \tag{4.36}$$

First test: machinery

In this section, only (4.33) is enforced, with $c_1, c_3 = 0$ in (2.16). The FEM domain of the problem is a cube 1m × 1m × 1m with a working frequency of 100 MHz, as shown in Fig 4.8, where also the analytical solution $\mathbf{E}_{\text{MMS}}$, FEM solution $\mathbf{E}_{\text{FEM}}$ and error $|\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}|$ are displayed. Values for both energy errors in a problem with 260 elements and



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

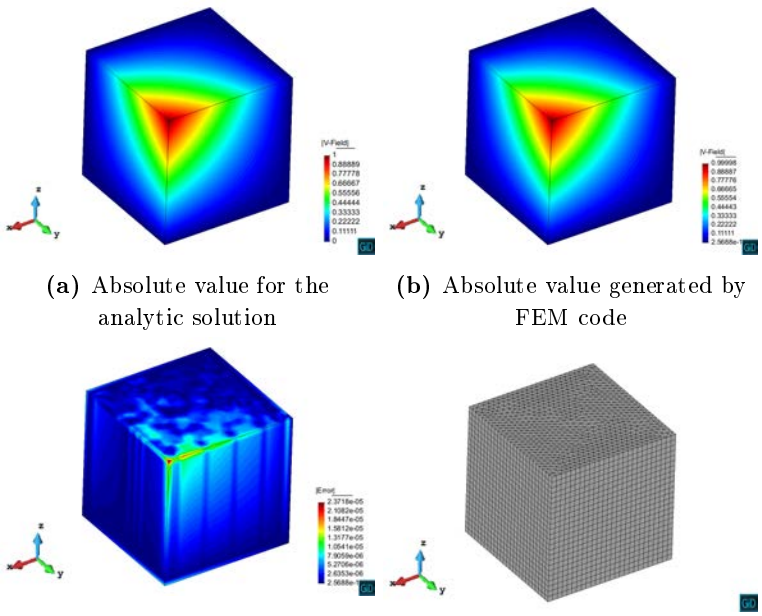
(d) Detail of the mesh

Figure 4.8: First MMS test using basis functions obtained with systematic approach applied to triangular prisms

4330 unknowns is $\varsigma = 0.3109\text{e}{-14}$ and $\varsigma_{\text{curl}} = 0.4006\text{e}{-13}$. In following subsections tests varying the size of the mesh are performed; here, since it is a debugging test to check the machinery, only results for one mesh are shown.

Second test: formulation

Next, every term in the formulation is checked varying the size of the problem under study, which is the same as in the previous section. Four different meshes have been generated as shown in Table 4.21 where results for the monomial in (4.36) are included. All the results are numerically low although the finer the discretization, the bigger the error. This is due to the accumulation of numerical error which is higher when more elements (which means more operations) are used. These results hold for all the 36 monomials which reside on the space of basis functions defined in (4.11). The last case in the Table 4.21 is displayed in Figure 4.9, where same representation as in Figure 4.8 is plotted. Note



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 4.9: Second MMS test using basis functions obtained with systematic approach applied to triangular prisms using a cube as domain to solve

that no red areas are present in that figure and that the error in the approximation of the field is of the same order in all the mesh.

Finally, a test with a curved element is included in the following to test the performance of the code with this kind of structures. A cylinder with radius of 0.2 m and height of 1 m is now the domain of the problem to be solved, and the size of elements tested are included in Table 4.22, where comparable results to straight elements are obtained. In Figure 4.10, the same set of figures as in Figure 4.8 and 4.9 is shown. Error is slightly bigger in the curved surfaces but bounded by the same order of magnitude found in Figure 4.9.

Third test: convergence

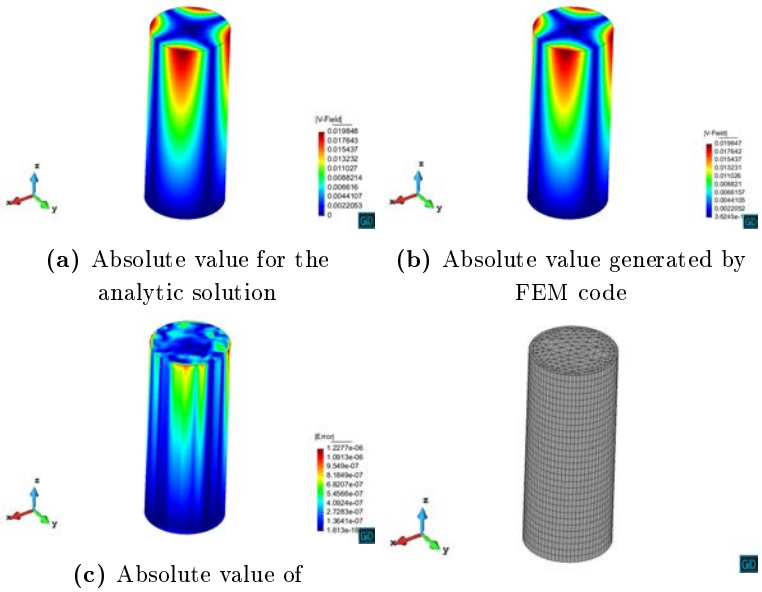
This last test consists of a convergence analysis with the smooth function introduced in (4.35). The same cube used in Section 4.7.1 has been used as problem to solve, and two polarizations, $\mathbf{E}_{\text{pol}} = \hat{\boldsymbol{\theta}}$ and $\mathbf{E}_{\text{pol}} = \hat{\boldsymbol{\phi}}$,

Table 4.21: Relative errors for the second test for triangular prisms using a cube as domain to solve

Elements	*Unknowns*	ς	ς_{curl}
16	340	0.869e−14	0.646e−14
260	4330	0.476e−13	0.111e−13
2080	31820	0.200e−12	0.204e−13
4272	64160	0.334e−12	0.253e−13

Table 4.22: Relative errors for the second test for triangular prisms using a cylinder as problem to solve

Elements	*Unknowns*	ς	ς_{curl}
48	910	0.128e−12	0.530e−14
80	1462	0.151e−12	0.627e−14
520	8544	0.537e−12	0.908e−14
1350	21576	0.935e−12	0.933e−14



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 4.10: Second MMS test using basis functions obtained with systematic approach applied to curved triangular prisms using a cylinder as problem to solve

and different angles of incidence have been considered. Results are shown in Figure 4.12 (where different incidence angles, different polarizations and different meshes with a frequency of 100 MHz are tested), and the convergence for each simulated excitation follow almost perfect straight lines. Different error levels are obtained depending on the polarization and angle of incidence with respect to the prisms. The relative position of the angle of incidence with the triangular faces of the prism is better detailed in Figure 4.11. Apart from this difference on the levels, in some specific cases the slope is even greater in absolute value than 2, which is the value predicted by the theory. Specifically, a sort of superconvergence behavior (slope equal to 3) is observed for an angle of incidence $\theta = 0°$ independently of the polarization, and for $\theta = 90°$ only with θ-polarization. By departing a few degrees from those specific angles the superconvergence behavior is lost and the slope is reverted to 2.

For comparison purposes with an unstructured space of basis functions, results with the tetrahedra of the same family of basis functions

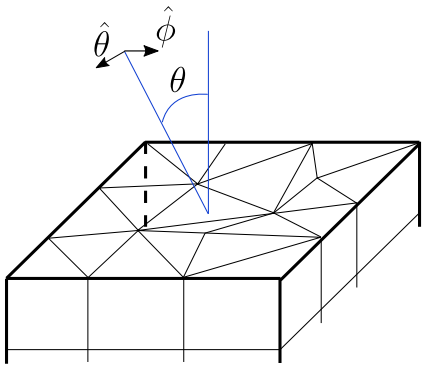


Figure 4.11: Detail of the mesh with triangular prisms including system of coordinates

are included in the figure. The error observed using this kind of unstructured elements is the same independently of the angle of incidence and polarization. This is due to the different construction of the spaces of functions: finite element space of the prism (4.11) is not symmetric/isotropic: there is a coordinate (local ζ coordinate of the prism) in which the approximation is different with respect to the other two coordinates. In contrast, tetrahedral elements expanding true Nédélec spaces (which are symmetric) approximate functions independently of the relative "orientation" of the mesh with respect to the solution coordinates.

Cavities

In this section, a number of cavity problems are analyzed to further validate the prismatic element with different materials and resonant structures, although the correctness of the basis functions and the machinery associated has already been tested with sections above. Variational formulation included in (2.16) is converted into a generalized eigenvalue problem setup with k_0^2 as eigenvalue by means of removing bilinear form $c_3(\mathbf{F}, \mathbf{V})$ and linear form $l(\mathbf{F})$.

Results of one empty (homogeneous) cavity and one non-homogenous cavity used before in the literature as benchmark structures have been selected. Specifically, the empty cavity has the shape of a triangular prism, with an equilateral triangular base of size 1 m and a height equal to the length of the triangular side (see [54]), with a working frequency of $f = 100$ MHz. Same working frequency is used for the non-homogeneous

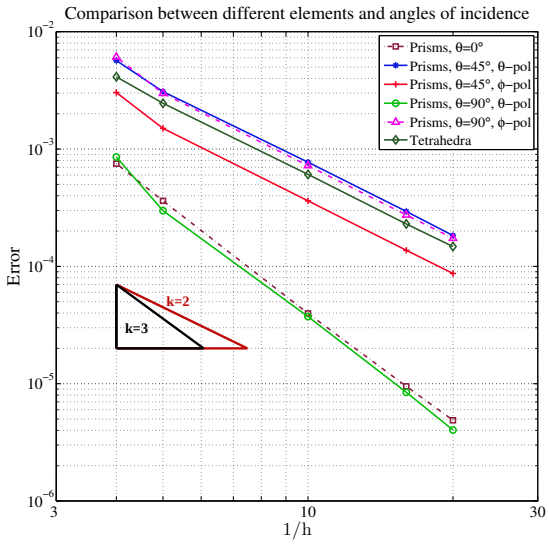


Figure 4.12: Convergence rate of the error over a cube using a smooth function as manufactured solution

cavity, formed by a metallic half-filled geometry of rectangular shape and dimensions 1m $\times$ 0.1m $\times$ 1m, with the upper half part of the third dimension filled with dielectric of $\varepsilon_r = 2$. Convergence rate of the average error in the computation of the first eigenvalues of these two cavities are included in Figure 4.13. In both cases, the obtained rate of convergence is very close to the expected theoretical rate of h^{2p}, [1], i.e., h^4 in our case.

4.7.2 Hierarchical tetrahedra

First and second test with MMS are applied to tetrahedra from [3] to check the correctness of the functions in this section. Since the strategy of assembly is different than the family obtained with the systematic approach, these tests have been a key factor to implement this kind of basis functions in the code.

First test: machinery

Same domain as in as in Section 4.7.1 is solved in this section, where the machinery of the code and the introduction of non-homogeneous Dirichlet BC explained in Section 4.6 are checked. The order of the basis

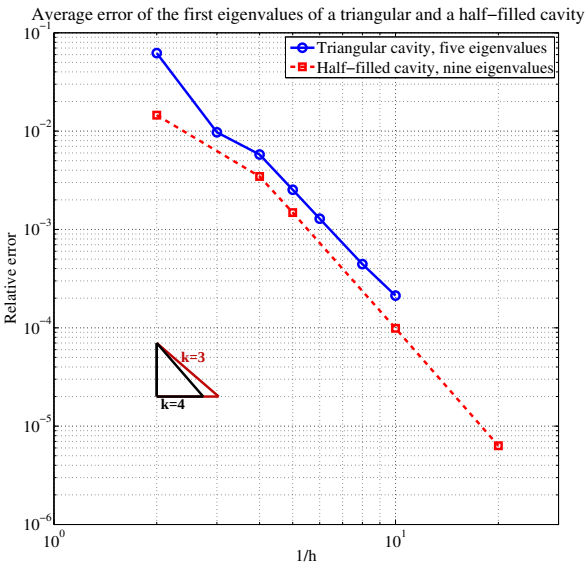


Figure 4.13: Convergence rate of the triangular prism with two different cavities

functions varies from order 2 to order 5 in order to check the machinery for different orders. The monomial from the space of functions with $p = 2$ chosen to show the results is

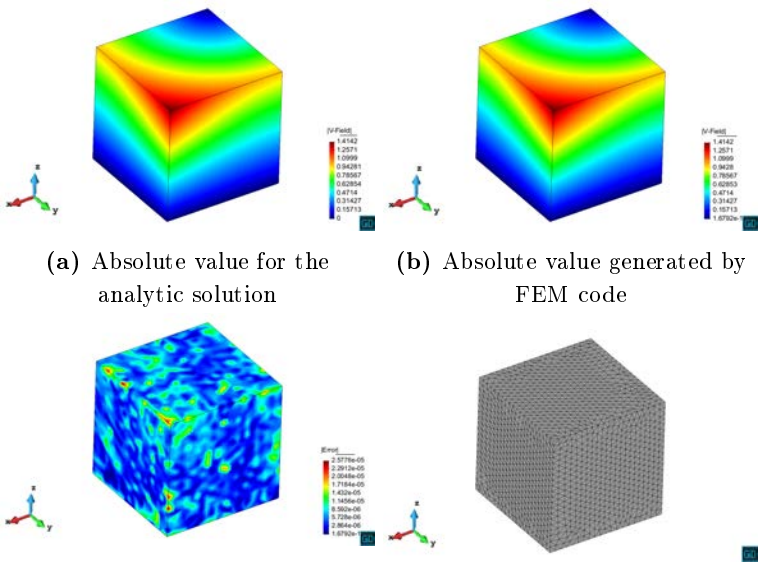
$$\mathbf{E}_{\text{MMS}} = \left\{ \begin{array}{c} yz \\ -xz \\ 0 \end{array} \right\}, \tag{4.37}$$

while for order $p = 5$ is

$$\mathbf{E}_{\text{MMS}} = \left\{ \begin{array}{c} -y^3z^2 \\ 0 \\ xy^3z \end{array} \right\}, \tag{4.38}$$

but, as in the sections above, all the 20 monomials for order $p = 2$ and 120 monomials for order $p = 5$ have been tested obtaining similar results.

In the Figure 4.14 the same set of figures shown in Figure 4.8 are included for order $p = 2$, and same comments can be extracted. Here,



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 4.14: First MMS test using hierarchical basis functions of order $p = 2$ applied to tetrahedra

higher error is not present in the points where the field has higher magnitude; this is due to the characteristic of the test since not all the machinery is tested. For the sake of brevity, and since similar conclusions can be extracted independently of the order used in the family of basis functions, results with order $p = 2$ are included for the first test while, for the second test, order $p = 5$ is used.

Regarding numerical results, for order $p = 2$ the numerical error obtained is, e.g., for monomial introduced in (4.37), $\varsigma = 0.5257\text{e}{-15}$ and $\varsigma_{\text{curl}} = 0.2547\text{e}{-14}$; while for order $p = 5$ and monomial defined in (4.38), $\varsigma = 0.1511\text{e}{-12}$ and $\varsigma_{\text{curl}} = 0.8709\text{e}{-12}$. An increase in the error is detected due to the numerical error which is increased with the growth of numerical operations.

Second test: formulation

Same test bench as in Section 4.7.1 is applied, with the same domain. All the 20 monomials belonging to the space of basis functions for $p = 2$ are tested but again, for brevity, only results corresponding to the same

Table 4.23: Relative errors for the second test for second order hierarchical vector basis functions in tetrahedra

Elements	*Unknowns*	ς	$\varsigma_{\mathrm{curl}}$
324	2340	0.101e−13	0.457e−14
1152	8112	0.191e−13	0.500e−14
2949	20200	0.343e−13	0.625e−14
9882	65814	0.851e−13	0.771e−14

Table 4.24: Relative errors for the second test for fifth order hierarchical vector basis functions in tetrahedra

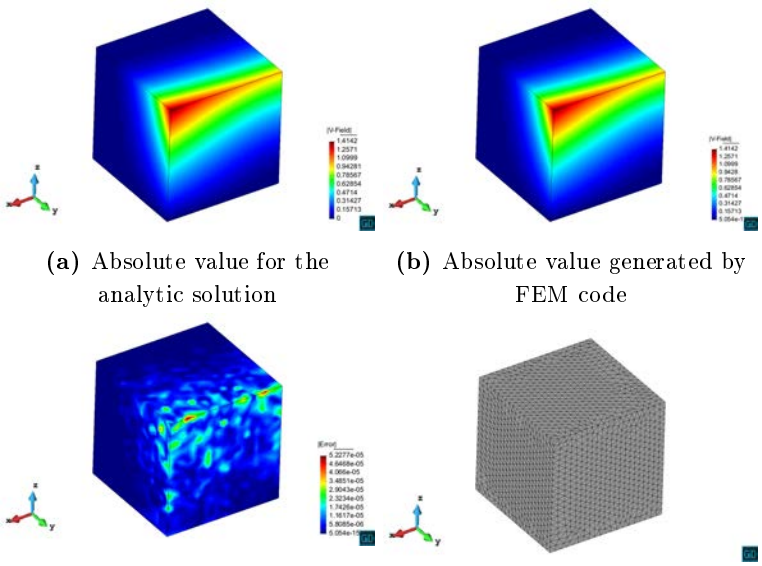
Elements	*Unknowns*	ς	$\varsigma_{\mathrm{curl}}$
324	2340	0.186e−11	0.264e−12
1152	91740	0.313e−11	0.273e−12
2949	231850	0.516e−11	0.274e−12
9882	766815	0.113e−10	0.304e−12

monomial of the previous section are detailed.

In Tables 4.23 and 4.24 results decreasing the size of the elements in the mesh are included for order $p = 2$ and $p = 5$ respectively. The same conclusions about the correctness of the basis functions and effect of numerical error on the results with a decrease on the size of the elements can be drawn. For the last case included in Table 4.24, the same set of results as in figures shown in previous sections is included, and apart from a lower level in the absolute error due to the increase in the order (which makes more appreciable the division into elements of the mesh), no further conclusions different from the already commented can be extracted.

4.7.3 Hierarchical hexahedra

Regarding hexahedra, only the second test is included in the following for brevity. For order $p = 2$, the monomial used in Table 4.25 is $\mathbf{E}_{\mathrm{MMS}} = (0, 0, x^2y^2z)$ but the whole set of 54 monomials belonging to the space of basis functions has been tested; while for higher orders, e.g., $p = 3$, the



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 4.15: Second MMS test using hierarchical basis functions of order $p = 5$ applied to tetrahedra

representative monomial used is $\mathbf{E}_{\text{MMS}} = (0, x^3y^2z^3, 0)$ in Table 4.26, although the 144 monomials have been checked. As in previous sections, the last case in Table 4.26 is used to show the results included in the Figure 4.16, where the same performance as with the other shapes is observed.

4.7.4 Hierarchical prisms

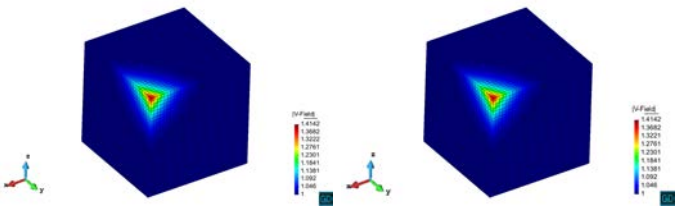
Finally, the second test is used to validate triangular prisms with hierarchical vector basis functions. Now, for order $p = 2$, the representative monomial used to obtain the results shown here is $\mathbf{E}_{\text{MMS}} = (xyz^2, -x^2z^2, 0)$ but any monomial out of the 36 which build the space of functions for the triangular prism would have generated the same kind of results. In Table 4.27 results increasing the number of elements are shown and the same performance as in previous sections has been obtained. Regarding higher orders, e.g., $p = 3$, the monomial used is $\mathbf{E}_{\text{MMS}} = (0, 0, x^3z^2)$ although the 90 monomials belonging to this space have been tested. In Table 4.28, the same test bench as in the other ta-

Table 4.25: Relative errors for the second test for hierarchical vector basis functions of $p = 2$ in hexahedra

Elements	*Unknowns*	ς	$\varsigma_{\mathrm{curl}}$
64	1944	0.237e−14	0.112e−13
512	13872	0.115e−13	0.238e−13
1000	26460	0.183e−13	0.316e−13
3375	86490	0.467e−13	0.814e−13

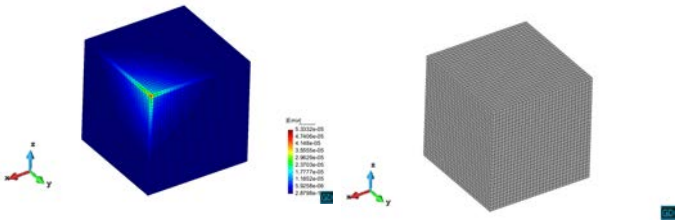
Table 4.26: Relative errors for the second test for hierarchical vector basis functions of $p = 3$ in hexahedra

Elements	*Unknowns*	ς	$\varsigma_{\mathrm{curl}}$
64	6084	0.681e−14	0.304e−13
512	45000	0.176e−13	0.533e−13
1000	86490	0.286e−13	0.946e−13
3375	285660	0.666e−13	0.201e−12



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code



(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 4.16: Second MMS test using hierarchical basis functions of order $p = 3$ applied to hexahedra

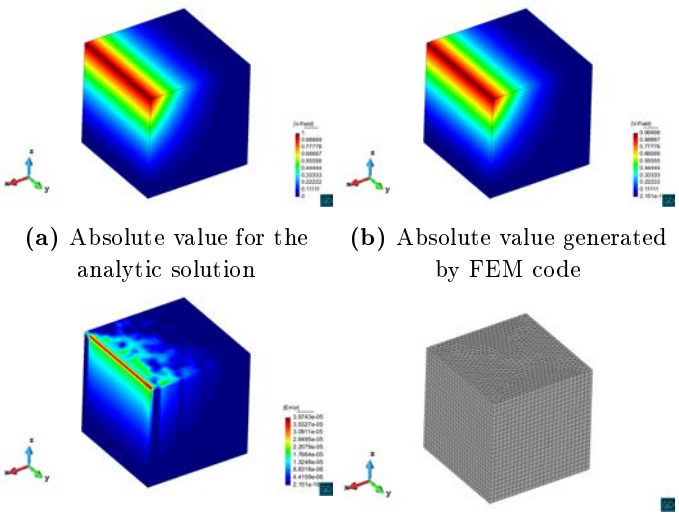
bles related to this section is applied, obtaining the same performance, and in Figure 4.17 the same set of results are included, leading to the same conclusions.

Table 4.27: Relative errors for hierarchical bases of order $p = 2$ applied to triangular prisms

Elements	*Unknowns*	ς	$\varsigma_{\mathrm{curl}}$
16	340	0.646e−14	0.354e−14
260	4330	0.107e−13	0.408e−14
2080	31820	0.355e−13	0.532e−14
4272	64160	0.600e−13	0.532e−14

Table 4.28: Relative errors for hierarchical bases of order $p = 3$ applied to triangular prisms

Elements	*Unknowns*	ς	$\varsigma_{\mathrm{curl}}$
16	966	0.340e−14	0.115e−13
260	13191	0.366e−13	0.107e−13
2080	99474	0.602e−13	0.119e−13
4272	201708	0.835e−13	0.127e−13



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 4.17: Second MMS test using hierarchical basis functions of order $p = 3$ applied to triangular prisms

4.8 Phase error

Regarding dispersion error, the use of unstructured meshes of tetrahedra is more advantageous in FEM code, [145], although all the shapes included in previous sections can be used. In this section, the three-dimensional shapes included in this dissertation are measured in terms of the dispersion error and its use in real elements is discussed, with special focus on the structured features of the triangular prism and hexahedron.

The dispersion error is closely related to the approximation of the phase magnitude of the solution which is not negligible for large scale problems. Thus, a phase error is obtained with MMS and results regarding the approximation of the phase in a long waveguide are included. Comparison between the three different elements are included in these two cases of study.

4.8.1 Phase error with MMS

The excitation of the problem to be solved is the exponential function defined in (4.35) while the domain problem is the same cube defined in Section 4.7.1 with the same working frequency of 100 MHz.

First, a comparison between triangular prisms and tetrahedra are included in Figure 4.18. Here, results corresponding to θ-polarization, i.e., $\mathbf{V}_{\text{pol}} = \hat{\theta}$, and different angles of incidence, are included. In the case of triangular prisms, note that this direction $\hat{\theta}$ is approximated by different components in the space of basis functions depending on the angle of incidence; e.g., for $\theta = 0°$, $\hat{\theta}$ is approximated by a linear combination of x and y components (associated with the triangle simplex space), while for $\theta = 90°$, $\hat{\theta}$ is approximated by z component, associated with the segment space. Remaining θ angles of incidence between $0°$ and $90°$ are linear combinations of the three Cartesian components.

The error shown in Figure 4.18, denoted as ς_{phase}, is computed as

$$\varsigma_{\text{phase}} = \frac{\displaystyle\int_{\Omega} \left| \measuredangle \mathbf{V}^{\theta}_{\text{FEM}} - \measuredangle \mathbf{V}^{\theta}_{\text{MMS}} \right| \, d\Omega}{\displaystyle\int_{\Omega} \left| \measuredangle \mathbf{V}^{\theta}_{\text{MMS}} \right| \, d\Omega}, \tag{4.39}$$

where $\measuredangle \mathbf{V}^{\theta}_{\text{FEM}}$ denotes the phase angle of the θ component of the field.

It is worth noting that dispersion error is not strictly represented by ς_{phase}, since dispersion error is the difference between the numerical and the exact values of the wavenumber (propagation constant).

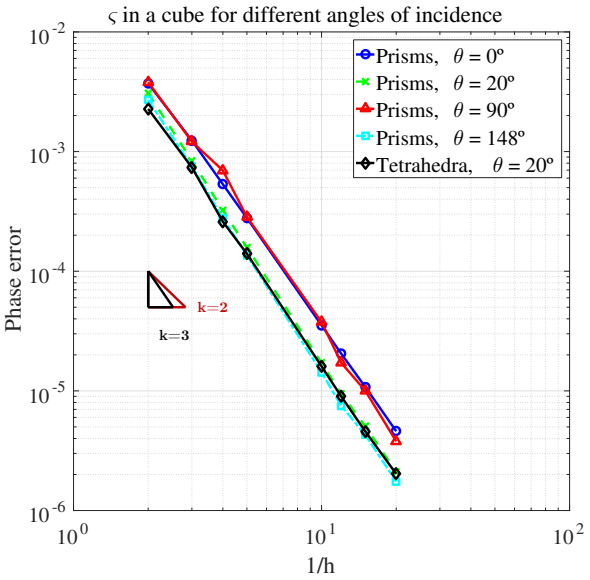


Figure 4.18: Convergence history of ς_{phase} defined in (4.39) using a complex exponential as manufactured solution with different angles of incidence

However, the observation of the behavior of ς_{phase} provides interesting conclusions. The first observation is that nice straight lines are obtained in all cases exhibiting an algebraic type of convergence. Furthermore, all lines present the same slope, which seems to be a value between 2 and 3. Note that theory predicts a slope equal to 2 for the second-order case and for the error in the whole field, and not only the phase error defined through (4.39). A second observation is the slight differences in the level of the error for the prism depending on the angle of incidence and, hence, on the field components excited. This is qualitatively the same behavior that is shown for the whole field in Figure 4.12; although no superconvergence is observed in the phase error. Similar results in directionality to the obtained with prisms can be found for hexahedra and they are not included in the graph for clarity purposes.

In order to further explore the non-symmetric features of the prism and the hexahedra spaces regarding the phase error, results of ς_{phase} obtained with a mesh of $h = 0.1\,\text{m}$ modifying the angle of incidence

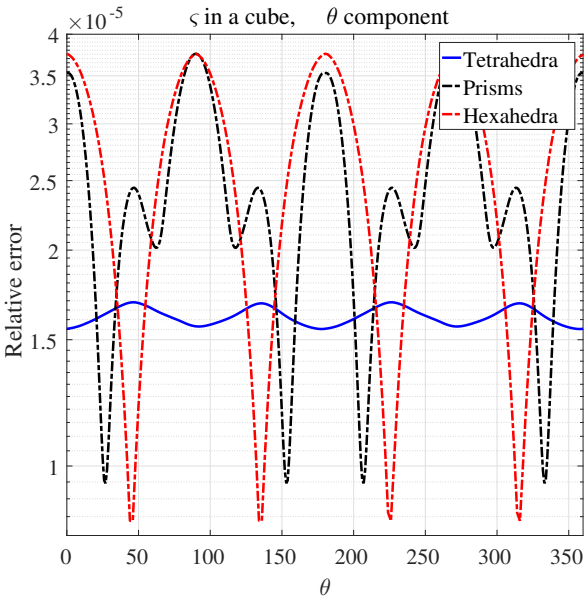
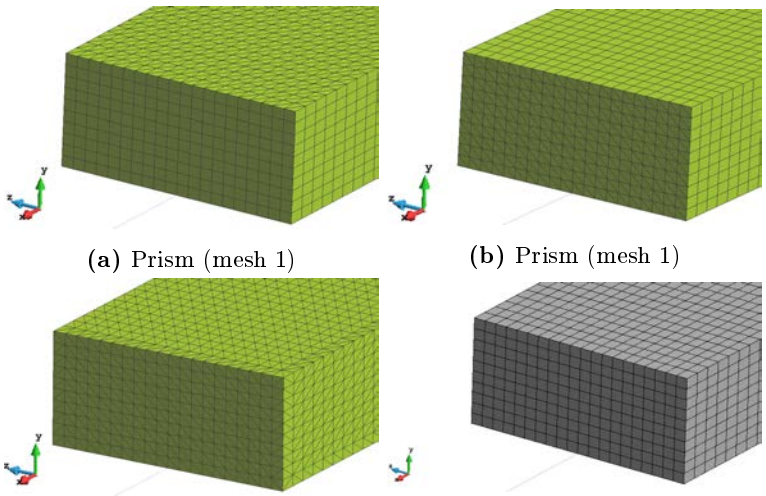


Figure 4.19: Behavior of ς_{phase} for a given mesh and different angles of incidence for the complex exponential as manufactured solution

θ are displayed together in Figure 4.19. The pattern generated with tetrahedral meshes is close to an omnidirectional case, while in the case of prismatic and hexahedral meshes the pattern clearly depends on the angle of incidence. Note that the directionality of the hexahedra follows an expected pattern: the variation from 0° to 45° is the same that the variation from 45° to 90° due to the construction of the space of functions for hexahedra.

4.8.2 Phase error in waveguides

Next, convergence behavior of the phase error obtained through the FEM analysis of a section of X-band rectangular WR-90 waveguide is included. The waveguide is excited with the fundamental mode at 7.5 GHz, and its length is 10λ at this frequency in a first simulation. Then, to make a comparison with higher orders, results with a length of 1λ are obtained. The phase error denoted here as Δk is obtained as $\Delta k = k_{\text{anal}} - k_{\text{FEM}}$, where k_{anal} is the analytic phase constant while k_{FEM} is the phase constant provided by FEM code computed through the phase of the S_{21}



(a) Prism (mesh 1)

(b) Prism (mesh 1)

(c) Tetrahedra

(d) Hexahedra

Figure 4.20: Different types of meshes of the rectangular waveguide

parameter between the two waveguide ports of the section. Four kinds of discretization are used (see Figure 4.20). Two types of discretization with prisms are considered: *Prism (mesh 1)* and *Prism (mesh 2)*. For *Prism (mesh 1)* the waveguide is tessellated with triangles lengthwise which are extruded along the transverse direction, meshing the waveport with quadrilaterals, while for *Prism (mesh 2)* the cross-section of the waveguide is meshed with triangles that are extruded lengthwise. The first option is the best option to discretize a waveguide with an arbitrary shape lengthwise, while the last option is the straightforward way in which an arbitrary cross-section waveguide would be meshed with prisms. Finally, the discretizations with hexahedra and unstructured tetrahedra have also been considered. The convergence of the phase error with respect to λ/h, where h is the size of the elements lengthwise, for the four types of discretization is displayed in Figure 4.21. The cross section is meshed with 20×10 elements in each case to have a fair comparison.

Next, results with higher polynomial orders for hierarchical vector basis functions are included. For computational reasons, now the length of the waveguide to be simulated is reduced to 1λ obtaining the same shape of results for order 2 (see Figure 4.22), where structured meshes

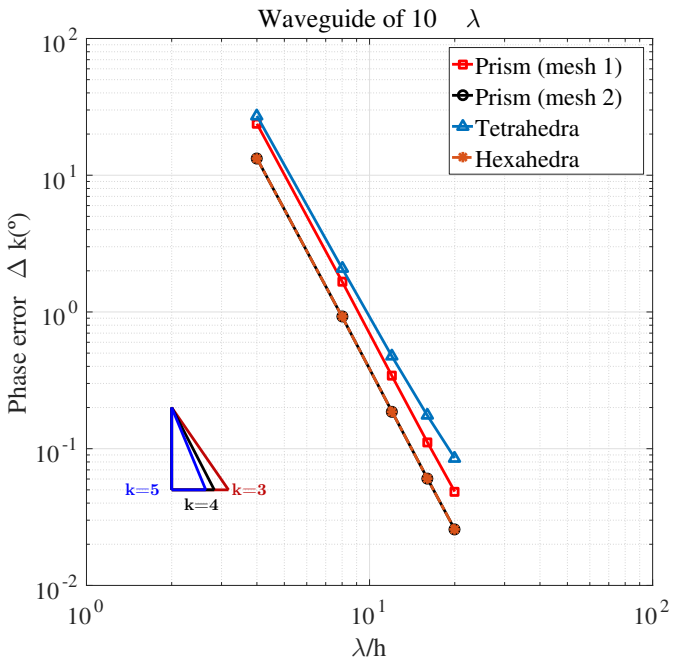


Figure 4.21: Convergence history of the phase error for the rectangular waveguide section

have been used for the four discretizations. Different polynomial orders, up to order $p = 5$, are applied to the same meshes obtaining the results included in Figure 4.23. The slopes obtained in this logarithmic figure are explicitly detailed in Table 4.29.

Note that the slopes obtained are close to those predicted by the theory included in [146]: error $= Ch^{2p}$. Each solution follows smooth straight lines. Moreover, the error obtained with *Prism (mesh 1)* overlaps with the error obtained with hexahedra as expected since the phase is approximated by the same space of functions (which is the polynomial space for the segment, while $\boldsymbol{\mathcal{R}_k}$ is used here to approximate the field on the transverse face). Nevertheless, *Prism (mesh 2)* is not exactly the same as tetrahedral discretization since the construction of the space of basis functions is not the same, i.e., the propagation direction is approximated by different spaces of functions obtaining different results. Also, note that same conclusions drawn from Figure 4.22 hold for higher

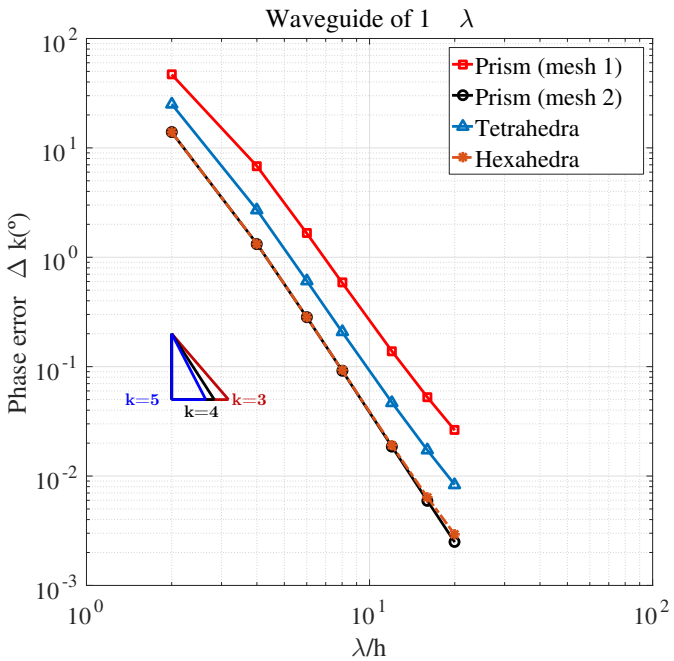


Figure 4.22: Convergence history of the phase error for the rectangular waveguide section with basis functions from $[1, 2]$

orders shown in Figure 4.23.

In [145], meshes in 2D with triangular faces are used to get the phase error, obtaining best results with unstructured or non-uniform meshes. This analysis is expanded to 3D getting better results with a non-uniform mesh composed of tetrahedra. However, in Figure 4.22 and 4.23, structured meshes have been used to perform a fair comparison between tetrahedra and semi-structured (prisms) or structured (hexahedra) meshes. In Table 4.30, some results with unstructured and structured meshes in one point of the graph where they can be compared (obtaining a mesh of 20×10 elements in the waveport for the unstructured case) are included. There, better results with unstructured meshes are obtained but with a similar order of magnitude. Finally, note that the discretization used in [145] to divide an hexahedron is tailored (5 tetrahedra for the non-uniform case and 6 tetrahedra for the uniform case) whereas the mesher used in these results, [147], takes up to 11 or 12 tetrahedra to discretize

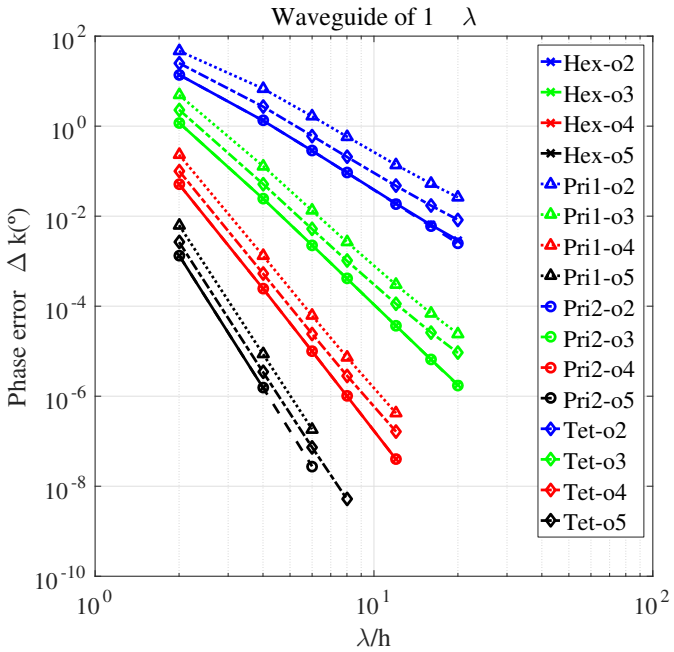


Figure 4.23: Convergence history of the phase error for the rectangular waveguide section with different element types from [3]

one hexahedron so the results obtained in [145] cannot be translated directly to the results presented here.

4.9 Conclusions

Two families of basis functions have been introduced in this chapter. The first family follows a systematic approach to obtain basis functions, providing stable and well-conditioned bases. On the other hand, the second family is based on [26] and it eases the introduction of non-uniform p refinement within the same domain. All the concepts needed to formulate the systematic basis functions has been introduced, from the space of functions to the DOFs. Also, different assembly strategies which arise from the discretization of the DOFs have been studied, showing in Sec 4.4.3 that the so-called *vq* strategy is better for assembly between triangular faces and that a better performance for assembly between rectangular faces is obtained by the so-called *vc* strategy due to the

Table 4.29: Slopes obtained experimentally with 1λ waveguide

	Theory	*Experimental value*			
Element type	All	Prism 1	Prism 2	Tetrahedra	Hexahedra
Order 2	4	2.917	3.600	3.128	2.895
Order 3	6	5.138	5.883	5.201	5.806
Order 4	8	7.368	7.885	7.419	7.887
Order 5	10	9.498	9.847	9.437	9.764

Table 4.30: Phase error $\Delta_k(^\circ)$ with structured and unstructured meshes

	Structured mesh	*Unstructured mesh*
Tetrahedra	9.596e-05	8.414e-05
Prism (mesh 1)	1.461e-03	4.526e-04

structured characteristic of the elements. Particularizing the systematic family of finite elements to triangular prisms (which can be understood as a hybrid between structured elements as hexahedra and simplices as tetrahedra), a detailed comparison with other families well-known in the literature is introduced in Section 4.5, obtaining comparable results.

Verification of the two families through MMS is introduced in Section 4.7, showing a detailed procedure of debugging basis functions that can be applied in general to other cases. Basically, if monomials which belong to the space of functions are taken as manufactured solutions, the solution provided by the code should be close to machine precision (apart from some accumulation of numerical error). On the other hand, if smooth functions such as complex exponential functions are under test, convergence results should follow straight lines with slopes predicted by the theory for all the three elements (tetrahedra, hexahedra, and triangular prisms) under test.

Finally, since a number of different shapes are being used in this dissertation, some experiments related to the phase error are included in the Section 4.8. The obtained results are promising, showing different performance in the structured elements depending on the angle of incidence of the wave propagation, but further studies increasing the electrical size

of the problem with the same mesh will follow in the future.

CHAPTER 5

VERIFICATION OF DDM

DDM is a technique introduced in electromagnetism in the last decade with many aspects to be considered and studied for its correct implementation. In this chapter, a procedure to verify and validate a DDM implementation introduced in a FEM code is thoroughly explained. This chapter is subdivided in four blocks:

- Verification of the formulation, introduced in Section 5.1.
- Study of the expected convergence of the iterative solver through the obtainment of the eigenspectra, detailed in Section 5.2.
- Verification of the implementation shown in Chapter 3, included in Section 5.3.
- Validation of DDM with some real problems and study of non-conformalities in a wave propagation problem, introduced in Section 5.4.

For the first and third block, MMS is used in a similar way as explained in Chapter 2. MMS is applied using a two-step procedure to isolate the different aspects involved in the introduction of DDM. First, the so-called *uncoupled* MMS is introduced, where cement variables are included as analytic solutions to check the formulation. Here, no machinery associated to the construction of the cross matrices is checked; but integration terms included in the formulation are verified. This part of the code is common to all the available shapes and orders, so only results with triangular prism with systematic space of basis functions defined in Section 4.2.3 are included.

Since the convergence of the iterative solver cannot be checked with MMS *a priori*, results showing the eigenspectra of the preconditioned matrix for the different TC are included with different combinations in the second block. This study makes possible to assure the convergence of subsequent results included in the third and fourth blocks.

After, MMS is applied as usual in formulations without DDM so all the machinery to create cross matrices and solve the surface problem with iterative solvers is checked. For this reason, a number of problems covering different combinations of shapes and orders is included in this block, and the effect of non-conformalities, hybrid and non-uniform p meshes is studied.

In the fourth block, a representative set of real problems are included to validate the technique. An expansion of the number of subdomains in one dimension (based on waveguides), two dimensions (using a square array of antennas) and three dimensions (through the computation of RCS of a plane) are included. Some performance results to check the right behavior of the implementation are also included.

5.1 Uncoupled MMS

Two new sets of variables have to be taken into account when applying MMS in DDM, corresponding to cement variables $\mathbf{j}_i$ and ρ_i. Two steps which compose an approach not documented before in the literature to debug DDM in electromagnetism are proposed: first, an *uncoupled* MMS is defined through the introduction of analytic $\mathbf{j}_i$ and ρ_i in the RHS without any assembly of the cross matrices; then, only the electric field $\mathbf{E}_i$ is introduced analytically to the problem whereas cement variables are approximated by the code. The first step is designed to test only the formulation, without the burden of checking the machinery to communicate subdomains (i.e., the use of shared points proposed in Chapter 3). The formulation and machinery are checked in the second step in the same way as MMS is used for *classic* FEM and results are included in Section 5.3, after the eigenspectra study. For some problems related to the implementation with scalar variables ρ_i, only developments with SOTC-TE are given in this section.

The formulation used in *uncoupled* MMS is introduced in the following. Let's consider the matrix equation for DDM in (2.38) and set cross matrices C_{ij} to zero, so communication with the other subdomains is

included in the RHS, i.e.,

$$\begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_n \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} \widehat{\mathbf{b}_1} \\ \widehat{\mathbf{b}_2} \\ \vdots \\ \widehat{\mathbf{b}_n} \end{pmatrix}, \tag{5.1}$$

where the new RHS $\widehat{\mathbf{b}_i}$ for each subdomain i is constructed by

$$\widehat{\mathbf{b}_i} = \begin{pmatrix} l(\mathbf{W}_i^i) \\ l(\mathbf{W}_i^s) \\ l_{\text{MMS}}(\mathbf{l}_i) \\ 0 \end{pmatrix}, \tag{5.2}$$

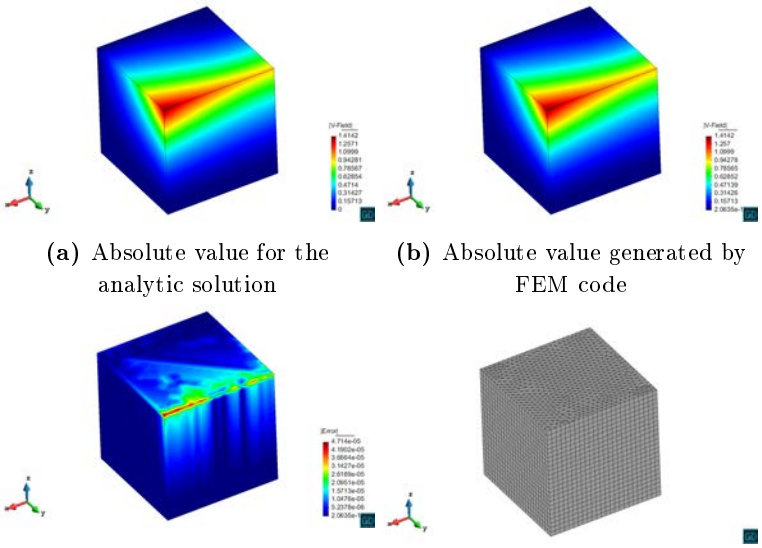
where $l(\mathbf{W}_i^i)$ is the linear form already defined in (2.37) and a new linear form is introduced here $l_{\text{MMS}}(\mathbf{l}_i)$ through

$$\begin{aligned} l_{\text{MMS}}(\mathbf{l}_i) = \alpha c_{\tau,1}(\mathbf{l}_i, \mathbf{e}_{\text{MMS}}) + \\ \beta_i k_0 c_{\tau,1}(\boldsymbol{\nabla}_\tau \times \mathbf{l}_i, \boldsymbol{\nabla}_\tau \times \mathbf{e}_i) + k_0 c_{\tau,1}(\mathbf{l}_i, \mathbf{j}_{\text{MMS}}). \end{aligned} \tag{5.3}$$

The bilinear form $c_{\tau,1}$ is defined in (2.37). In (5.3), different signs are found for terms related to $\pi_\tau^\times$ since the normal used is the normal for each subdomain, and not the normal from the neighbor subdomain, which only differs in the sign.

The triangular prism derived from the systematic approach is chosen to test the formulation due to its hybrid characteristic which means that both kinds of faces (triangular and rectangular) present in all the shapes are tested in the integration terms. For brevity purposes, only results with one monomial (defined in (4.36)) is included next, although each monomial included in the space of functions has been checked. However, in Section 5.3, all the finite element shapes are used to test both machinery and formulation through the second step of verification with MMS.

The domain under test is a cube of dimensions $1\,\text{m} \times 1\,\text{m} \times 1\,\text{m}$ where 2 subdomains have been defined through an interface set by the user (input data) in the diagonal along the top face. The working frequency is set to 100 MHz in all the MMS problems shown in this chapter. A matching and conformal mesh is used on the interface between subdomains, and errors defined in (4.31) and (4.32) are used. In Table 5.1



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 5.1: Uncoupled MMS with systematic triangular prisms applied to a cube

different meshes decreasing the size of the elements are generated and the same phenomena as in Section 4.7.1 is observed: the numerical error is accumulated but still kept under control. Note that the iterative solver is not used yet since the accuracy of the iterative solver might mask errors in the formulation. Moreover, in Figure 5.1 the analytical solution provided to the code, the solution generated by the code, the absolute error between those two and the detail of the mesh used (last case in Table 5.1) are shown. The division between subdomains can be observed in the Figure 5.1, where a slight increase in the error is present on the boundary between subdomains. However, the red area is present where the field is stronger, so in order to discard errors only due to the division into subdomains, a new simulation with the approximation of zero order monomial $\mathbf{E}_{\text{MMS}} = (1, 1, 1)$ is tested in the Figure 5.2, where no red areas are detected on the interface between subdomains.

Three parts of the code are tested through the following problem: division in more than two subdomains, use of ParMETIS to obtain different subdomains and curved structures in the outer boundary. Now, the

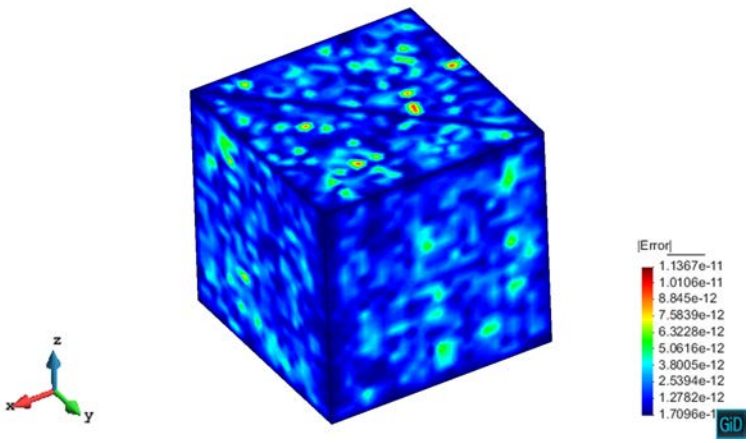


Figure 5.2: Absolute value of $\mathbf{E}_{\mathrm{MMS}} - \mathbf{E}_{\mathrm{FEM}}$ for uncoupled MMS with systematic triangular prisms approximating zero order monomials

Table 5.1: Relative errors for the uncoupled MMS for triangular prisms

Elements	*Unknowns*	ς	$\varsigma_{\mathrm{curl}}$
144	3446	0.187e−12	0.137e−13
270	5722	0.261e−12	0.178e−13
2080	36542	0.111e−11	0.287e−13
3864	64902	0.183e−11	0.366e−13

Table 5.2: Relative errors for the uncoupled MMS for triangular prisms using a cylinder as problem to solve

Elements	*Unknowns*	ς	ς_{curl}	Overhead (%)
48	1552	0.221e−12	0.106e−13	70.5%
80	2256	0.479e−12	0.133e−13	54.3%
520	10844	0.986e−12	0.142e−13	26.9%
1350	26274	0.141e−11	0.163e−13	21.8%

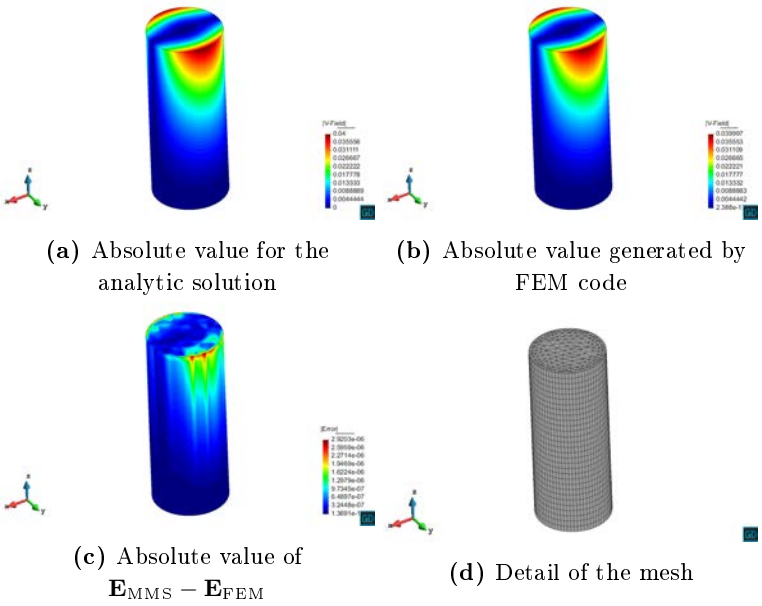
problem under test is the cylinder already used in Section 4.7.1 where ParMETIS is configured to generate 4 subdomains, same meshes included in Table 4.22 are used, and monomial $\mathbf{E}_{\text{MMS}} = (y^2z^2, -xyz^2, 0)$ is introduced. In Table 5.2 error energy results for the problem are included increasing the number of elements in the mesh. Moreover, to better understand the effect in the number of unknowns of the introduction of DDM, a measure of the so-called here *overhead* for using DDM (due to the inclusion of cement variables) is computed and included in Table 5.2. This overhead is defined as

$$\text{Overhead}(\%) = \frac{n_{\text{unkn, DDM}} - n_{\text{unkn, no DDM}}}{n_{\text{unkn, no DDM}}} \times 100, \tag{5.4}$$

where $n_{\text{unkn, no DDM}}$ and $n_{\text{unkn, DDM}}$ stand for the original problem without any call to ParMETIS and the number of unknowns of the DDM problem, respectively. In the following, to ensure a fair comparison, the overhead is only included in results when ParMETIS is called since the same mesh with and without DDM is compared. The magnitude of the errors are as expected, which means that the accuracy of the approximation is not affected by the introduction of DDM formulation, while the overhead is decreased with the size of the mesh since the number of subdomains is the same for all cases. Note that the overhead is not negligible, especially for smallest meshes: however, real problems are bigger and this overhead is not a problem as it is appreciated in Section 5.4. Finally, in Figure 5.3 results for the last mesh are shown, where no difference between subdomains is appreciated.

5.2 Study of the eigenspectra

As explained in Section 2.2, the introduction of DDM involves solving a surface problem for which an iterative solver is commonly used.



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 5.3: Uncoupled MMS with systematic triangular prisms applied to a cylinder

Due to the characteristics of iterative solvers, the convergence of the preconditioned problem included in (2.45) has to be checked for the different possibilities relative to shapes and orders available in the code. For this reason, the eigenspectra as introduced in [96] is studied in this section. In short, FEM matrix A obtained with DDM included in (2.38) can be decomposed into two matrices M, resulting from block-Jacobi preconditioning, and N, i.e.,

$$A = M - N; M = \begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_n \end{pmatrix}; N = \begin{pmatrix} 0 & -C_{12} & \dots & -C_{1n} \\ -C_{21} & 0 & \dots & -C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -C_{n1} & -C_{n2} & \dots & 0 \end{pmatrix}; \tag{5.5}$$

so (2.45) can be expressed as

$$M^{-1}A = M^{-1}(M - N) = I - M^{-1}N, \tag{5.6}$$

and the eigenspectra will be centered into the point $(1.0, 0.0)$ in the complex plane. The convergence of the iterative solver is assured if each eigenvalue is constrained into the unit circle centered in $(1.0, 0.0)$. If some eigenvalues are outside this circle, the iterative solver can converge if those eigenvalues are not excited by the electromagnetic problem. Besides, the convergence is faster with a more clustered set of eigenvalues around the center which can be achieved with different TC.

In the following, a representative set of cases which covers all the possible combinations in the code is included. The objective is to test *i)* problems fully meshed with all the available shapes (triangular prisms, tetrahedra and hexahedra); *ii)* the influence of different orders in the different subdomains; *iii)* division into subdomains through input data and through ParMETIS (which generates non-planar interfaces for tetrahedra and triangular prisms); *iv)* the effect of different elements for each subdomain with the same mesh on the interface; *v)* the influence of non-conformal interfaces between subdomains; and *vi)* the effect of Dirichlet BC which appear in, e.g., waveguides.

First, systematic vector basis functions in triangular prismatic elements to solve the problem shown in Figure 5.4, where a conformal division into two subdomains is introduced through input data. The eigenspectra generated for FOTC and SOTC-TE is shown in Figure 5.5, where a smooth exponential function has been used as excitation although the eigenspectra is not altered by the excitation (imposed in the RHS).

In the following, hexahedra with hierarchical vector basis functions of order $p = 3$ are used to discretize the FEM formulation to generate the eigenspectra included in Figure 5.7. A conformal division into 2 subdomains is introduced in the problem shown in Figure 5.6 through ParMETIS although the division between subdomains is planar since hexahedra are used.

Last shape to be tested is tetrahedra so the domain included in Figure 5.8 is divided with ParMETIS into two subdomains and the electric field is approximated with order $p = 4$, generating the eigenspectra shown in Figure 5.9. Note that in this case the interface between subdomains is non-planar and conformal.

Same characteristics are shared by all the eigenspectra shown so far: each eigenvalue is inside the unit circle so convergence of the iterative solver is assured for each possible excitation of the problem. Moreover,

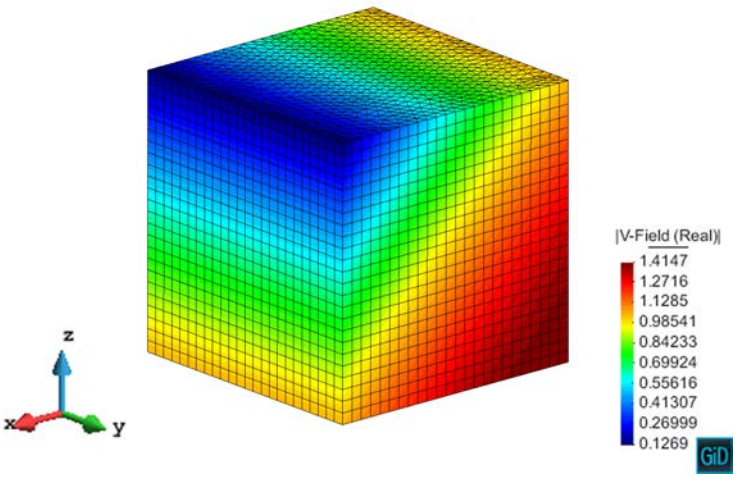


Figure 5.4: Mesh using a planewave as excitation to generate the eigenspectra for triangular prisms with systematic basis functions

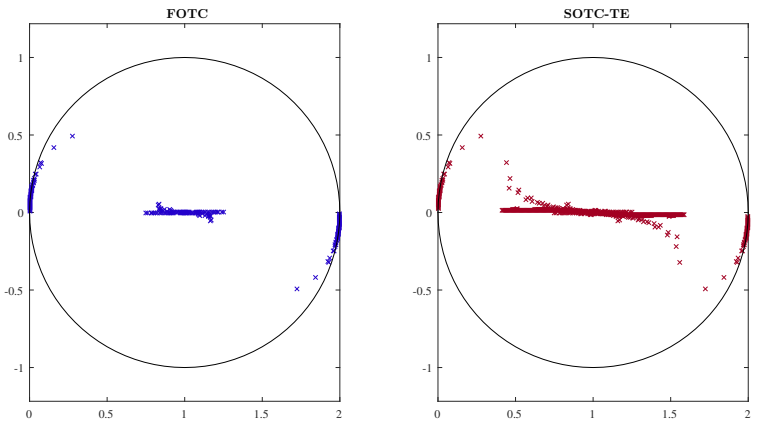


Figure 5.5: Eigenspectra from problem shown in Figure 5.4

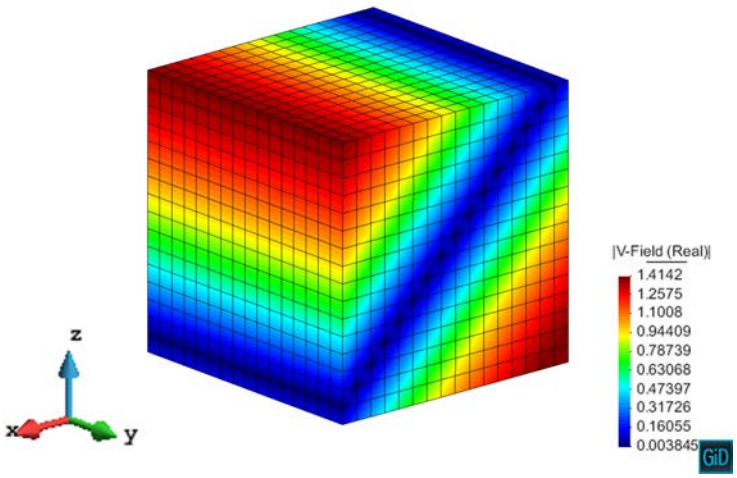


Figure 5.6: Mesh using a planewave as excitation to generate the eigenspectra for hexahedra

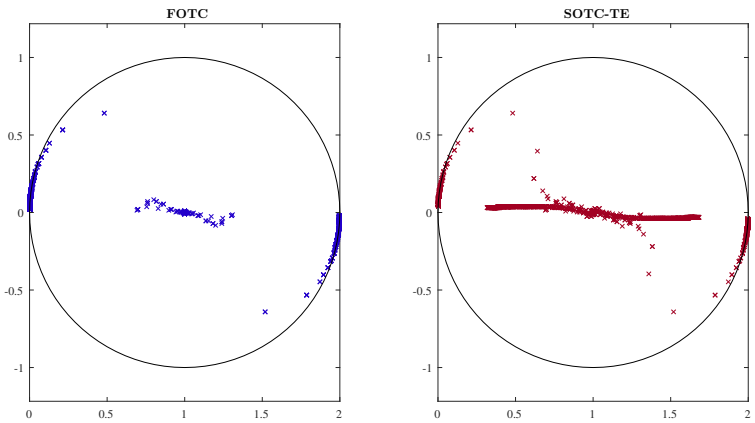


Figure 5.7: Eigenspectra from problem shown in Figure 5.6

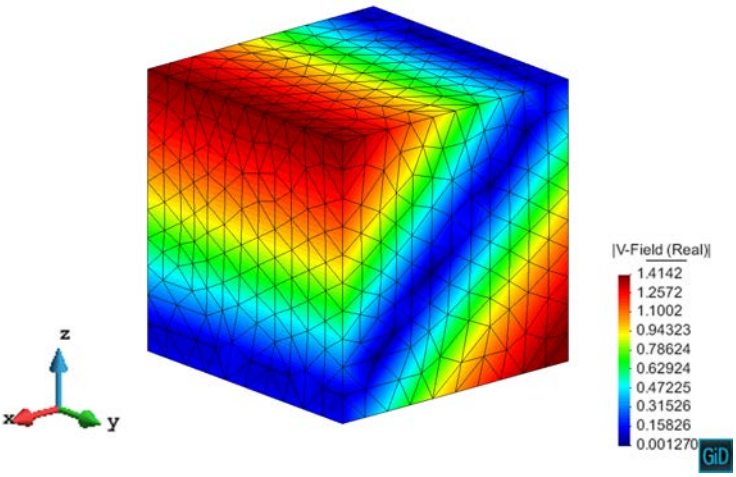


Figure 5.8: Mesh using a planewave as excitation to generate the eigenspectra for tetrahedra

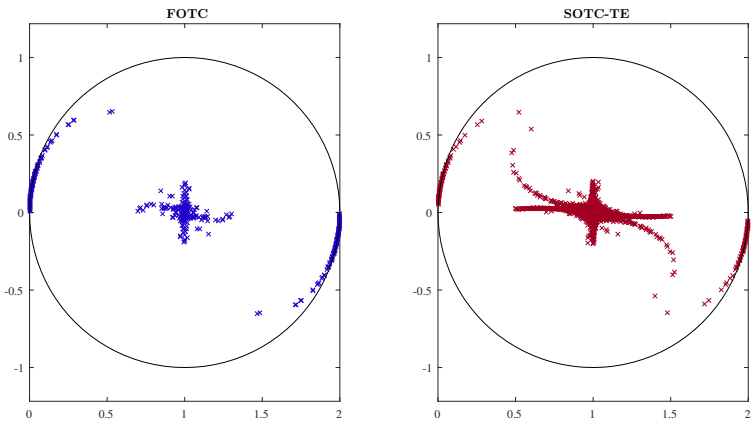


Figure 5.9: Eigenspectra from problem shown in Figure 5.8

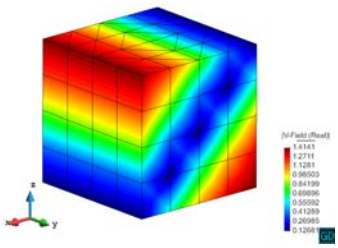


Figure 5.10: Mesh using a planewave as excitation to generate the eigenspectra for triangular prisms with $p = 4$

eigenvalues are more clustered around the center in the case of SOTC, where approximation of evanescent TE fields are included in the TC. No special differences are detected using different shapes and families of basis functions.

Now, a conformal division into 4 subdomains through ParMETIS is introduced in the problem included in Figure 5.10, where hierarchical vector basis functions of order $p = 4$ are used to approximate the electric field. Note that here the interface between subdomains is non-planar and the problem of corner edge is present, [96]. This problem is unavoidable to face (through some penalty term or other options) in non-conformal division; however, in this case, it can be seen in the eigenspectra shown in Figure 5.11 that the preconditioned eigenspectra has nonzero null eigenspace but since the excitation never belongs to that nullspace, the solution can be obtained. Note that, in this case, a direct solver cannot be used since the matrix is singular.

One of the remaining points to be tested is the division into subdomains discretized with different elements and matching mesh on the interface. For this specific case, a problem with two subdomains, where one subdomain is fully meshed with hexahedra and the other is meshed with triangular prisms, are created as shown in Figure 5.12. This problem is simulated generating the eigenspectra of Figure 5.13, where hierarchical vector basis functions of order $p = 3$ have been used for both shapes. A similar problem with a mix of tetrahedra and triangular prisms with triangular faces on the common interface between subdomains (see Figure 5.14) is simulated with hierarchical vector basis functions of order $p = 4$, obtaining the eigenspectra shown in Figure 5.15. No abnormali-

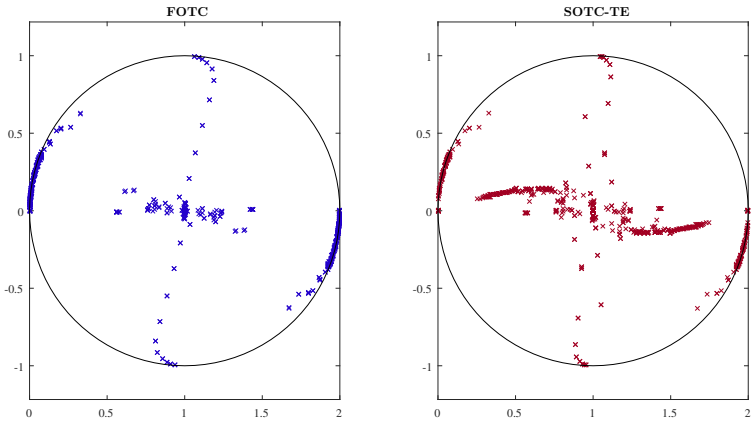


Figure 5.11: Eigenspectra from problem shown in Figure 5.10

ties in the eigenspectra are detected, so the convergence of the iterative solver is not affected by the shape used (even when different shapes are used in different subdomains).

The effect of non-conformality between subdomains in the sense of non-matching meshes on the interface is tested next. First, the problem shown in Figure 5.16 is simulated where tetrahedra are used to discretize each one of the two subdomains, and a non-conformal mesh is used on the interface. The eigenspectra obtained is shown in Figure 5.17, where systematic basis functions are used to approximate the solution. Moreover, to observe the effect on the convergence of the iterative solver, a mix between tetrahedra and triangular prisms is simulated as shown in Figure 5.18, where hierarchical vector basis functions have been used with different orders for each subdomain (order $p = 2$ for tetrahedra and order $p = 3$ for triangular prisms). The eigenspectra obtained is shown in Figure 5.19. In these two cases, more sparsity in the eigenvalue distribution is appreciated but same properties in terms of convergence of the iterative solver hold, even for different orders of approximation in the different subdomains.

Finally, the effect of Dirichlet BC is tested with a WR-90 waveguide where hierarchical vector basis functions of order $p = 3$ have been used to approximate the electric field and each subdomain is discretized with hexahedra, as shown in Figure 5.20. The working frequency is set to 10 GHz and the length l of the structure is set to $l = \lambda_g$, where λ_g is

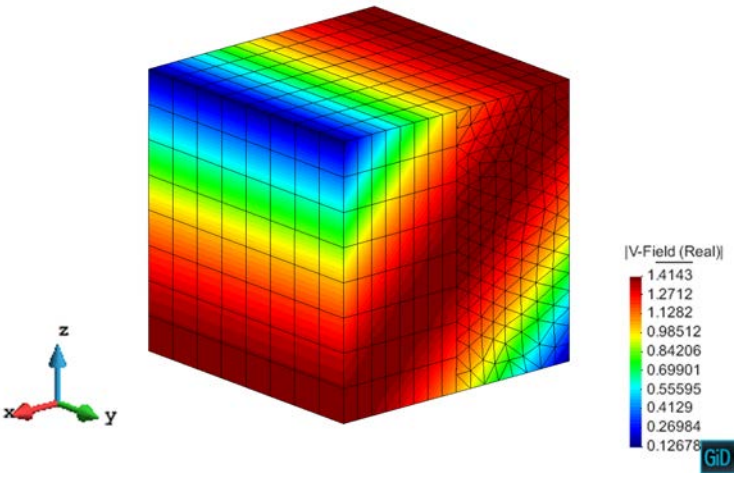


Figure 5.12: Mesh using a planewave as excitation to generate the eigenspectra for hybrid meshes between prisms and hexahedra

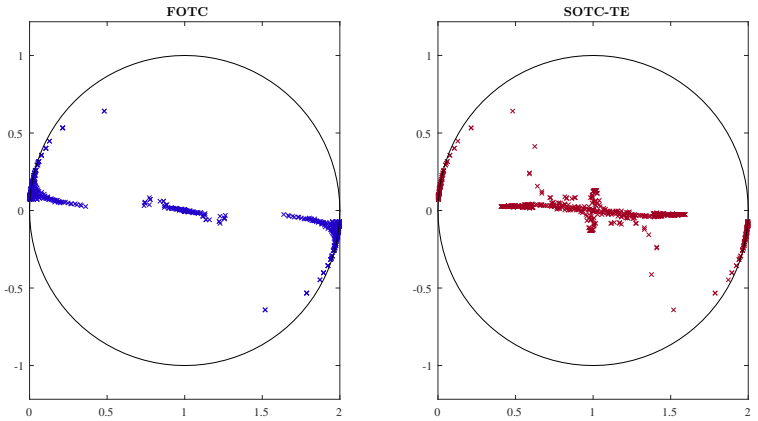


Figure 5.13: Eigenspectra from problem shown in Figure 5.12

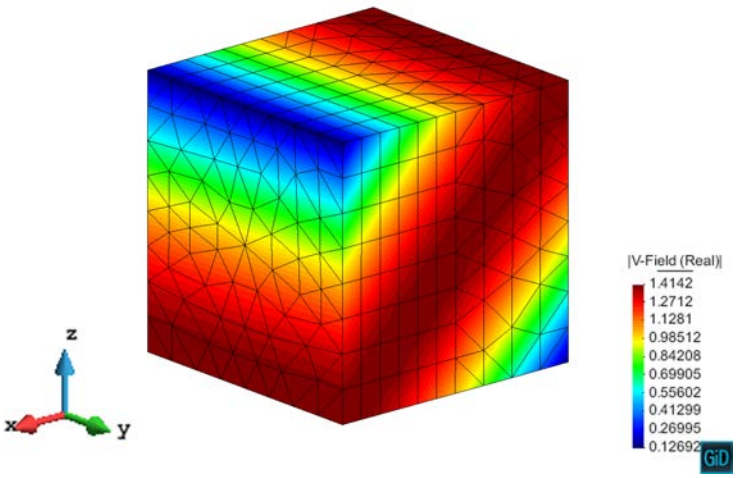


Figure 5.14: Mesh using a planewave as excitation to generate the eigenspectra for hybrid meshes between tetrahedra and triangular prisms

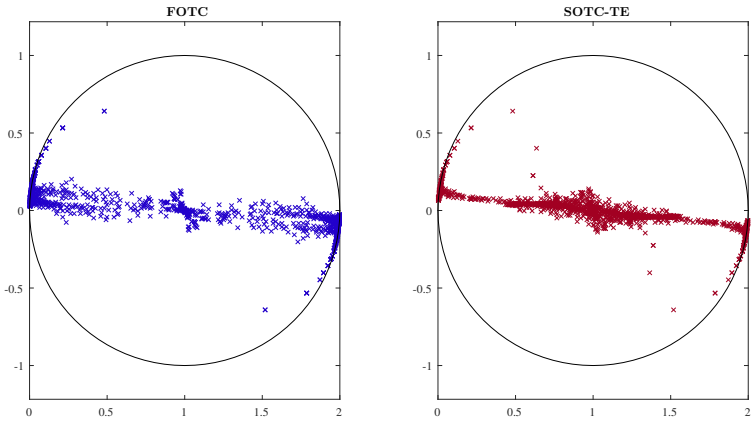


Figure 5.15: Eigenspectra from problem shown in Figure 5.14

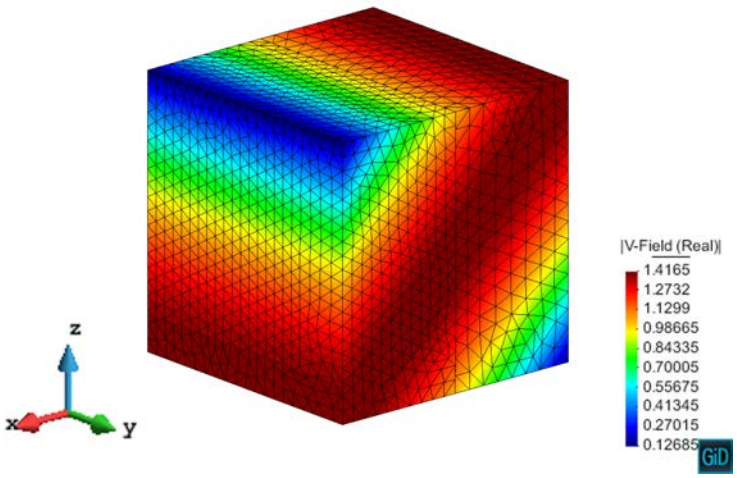


Figure 5.16: Mesh using a planewave as excitation to generate the eigenspectra for a non-conformal division of the problem with tetrahedra

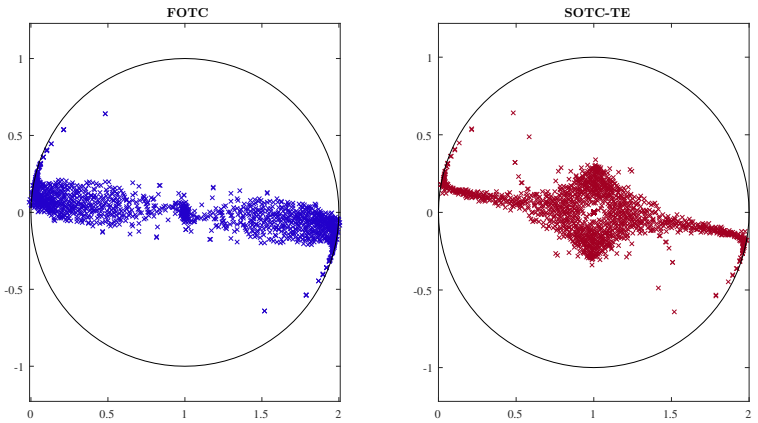


Figure 5.17: Eigenspectra from problem shown in Figure 5.16

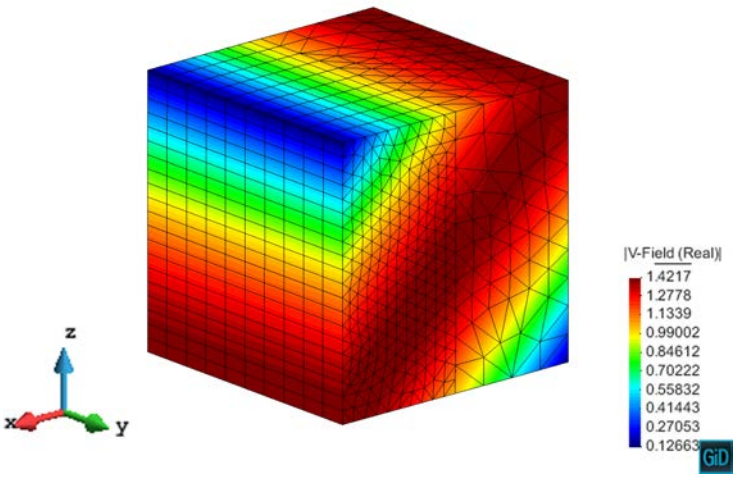


Figure 5.18: Mesh using a planewave as excitation to generate the eigenspectra for a non-conformal division of the problem with tetrahedra ($p = 2$) and prisms ($p = 3$)

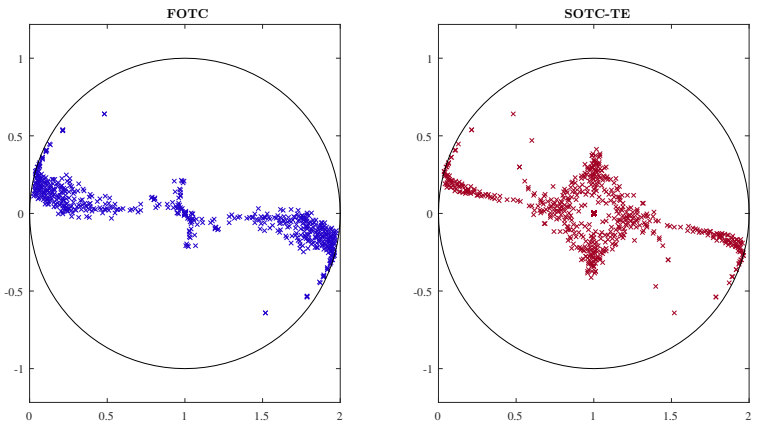


Figure 5.19: Eigenspectra from problem shown in Figure 5.18

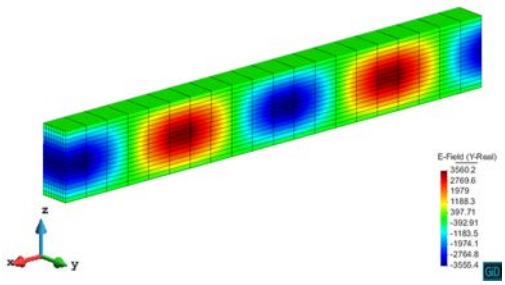


Figure 5.20: Mesh using hexahedra in a WR-90 waveguide

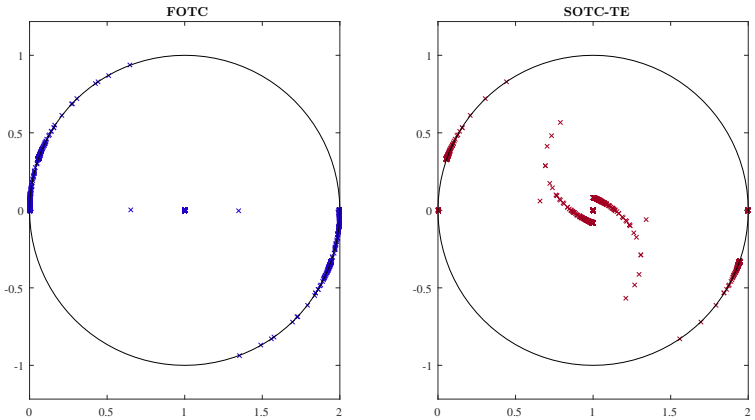


Figure 5.21: Eigenspectra from problem shown in Figure 5.20

the wavelength within the guided structure. For clarity purposes, when λ is used for waveguides, it means λ_g. A conformal division along the transversal face is performed through input data generating two sections of 0.5λ, obtaining the eigenspectra shown in Figure 5.21, where the eigenvalue shown in the center correspond to the trivial solution and the propagating modes can be shown for FOTC in the x-axis. Evanescent TE modes are displaced from the circumference to the center in the two braces shown for SOTC-TE transmission conditions, while the remaining evanescent TM modes are closer to the circumference. More details about the physical meaning of the eigenspectra are found in [91].

To sum up, no abnormalities have been found in any combination

studied in this section, in which a representative set of possibilities has been studied. The convergence of the iterative solver is expected to be better for SOTC-TE since a higher number of eigenvalues are clustered around the center of the circle.

5.3 Verification with MMS

So far, DDM formulation introduced in the code and the convergence of the iterative solver with a number of possible combinations has been tested. In this section, the same scheme followed in Section 4.7 to verify FEM formulation is included here for the DDM formulation defined in (2.38). Now, the special treatment to cement variables (even knowing its analytical solution) is not applied, so machinery to build cross matrices is tested since the formulation itself has already been checked in Section 5.1. PETSc iterative solver using generalized conjugate residual (GCR) method with a relative tolerance of 1e−12 has been introduced for checking possible stagnation in the results. FOTC and SOTC-TE are applied for comparison purposes. A representative set of combinations is included to cover all the possible situations, as in Section 5.2: *i)* all the shapes available; *ii)* higher orders; *iii)* use of ParMETIS; *iv)* hybrid meshes between elements with a matching mesh between subdomains; *v)* non-conformality between subdomains.

First shapes to be checked are tetrahedra using the systematic family of vector basis functions introduced in Section 4.2.1. The domain to be solved is a cube of dimensions $2\,\mathrm{m} \times 1\,\mathrm{m} \times 1\,\mathrm{m}$ which has been split by the user (through input data) in two halves, obtaining two identical cubes of dimensions $1\,\mathrm{m} \times 1\,\mathrm{m} \times 1\,\mathrm{m}$, constituting one subdomain each. The monomial used is $\mathbf{E}_{\mathrm{MMS}} = (xy, -x^2, 0)$ although the same kinds of results are obtained with the other monomials within the space of functions for tetrahedra. Results with an increase in the number of unknowns are included in Table 5.3. A better convergence is achieved with SOTC-TE as expected, with a higher number of iterations for problems with a higher number of elements, but no stagnation occurs in any case. Error results included here are limited by the accuracy of the iterative solver, so no effect related to the numerical error present in Section 4.7 is observed. In order not to mask results obtained with the accuracy of the iterative solver, a direct solver (MUMPS) is used to solve the global surface problem, generating results included in Table 5.4. The same phenomena related to the numerical error is obtained while the solution obtained is not affected by the user-driven division into subdomains.

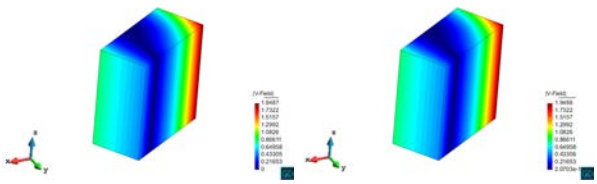
In Figure 5.22, the same set of results as in Figure 5.3 is included using the last mesh shown in Table 5.3. It is observed (as expected since the formulation has been tested and shown in Figure 5.2) that the matching division between subdomains is not a source of error when us-

Table 5.3: Relative errors for systematic tetrahedra using a cube split in two domains with an iterative solver

Elements	*Unknowns*	*Iterative solver*		Iterations	Iterations
		ς	ς_{curl}	(FOTC)	(SOTC-TE)
26	302	0.333e−11	0.102e−11	29	20
274	2378	0.266e−10	0.214e−10	94	43
5321	38394	0.118e−10	0.218e−10	147	59
11532	80860	0.110e−10	0.221e−10	161	67

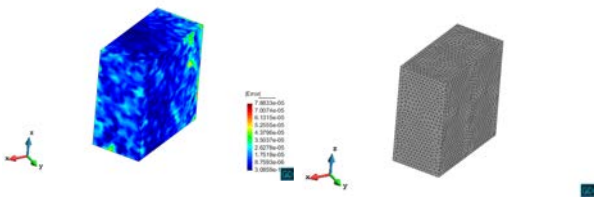
Table 5.4: Relative errors for systematic tetrahedra using a cube split in two domains with a direct solver applied to the surface problem

Elements	*Unknowns*	*Direct solver*	
		ς	ς_{curl}
26	302	0.295e−13	0.118e−13
274	2378	0.932e−13	0.194e−13
5321	38394	0.574e−12	0.301e−13
11532	80860	0.594e−11	0.219e−12



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code



(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 5.22: Results of MMS test with a monomial belonging to the space of functions of tetrahedra of order $p = 2$ with a cube as domain to solve

ing an iterative solver and, from the accuracy point of view, the division into subdomains is transparent. To avoid any mask of the error due to the use of the iterative solver, in Figure 5.23 the absolute error obtained through a direct solver is included, holding the same conclusions extracted before.

Now, in order to check the use of ParMETIS to generate the division into subdomains, 4 subdomains are generated automatically to solve the same problem introduced in Section 4.7.2. The monomial used to compute solutions is $\mathbf{E}_{\text{MMS}} = (xy, 0, 0)$, and hierarchical vector basis functions of order $p = 4$ are used so higher-order vector basis functions are also checked. In Table 5.5, results from the direct solver and the number of iterations needed to achieve convergence with different TC are obtained. The trend on the results is similar to Tables 5.3 and 5.4 with a higher numerical error than in Section 4.7.2. This increase in the numerical error is attributed to the machinery of DDM, which is much more complex numerically speaking that *classic* FEM and more operations are needed to obtain the solution in the whole domain. Finally, in Figure 5.24 results with the last mesh included in Table 5.5 are shown.

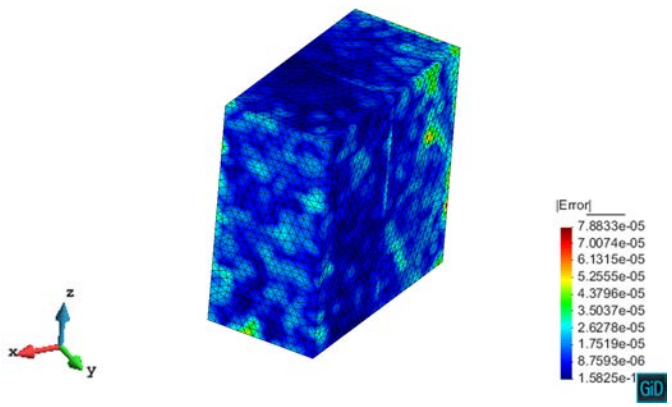


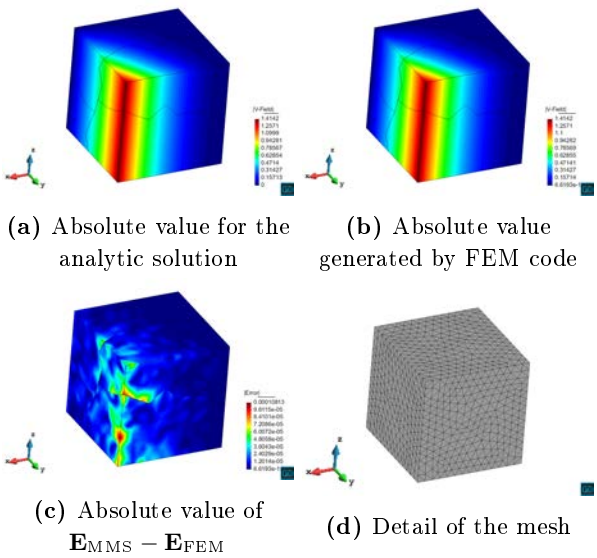
Figure 5.23: Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$ for the same problem as in Figure 5.22 using a direct solver for the surface problem

Table 5.5: Relative errors for hierarchical tetrahedra ($p = 4$) using 4 subdomains through METIS

Elements	*Unknowns*	*Direct solver*		Iterative solver	
		ς	ς_{curl}	(FOTC)	(SOTC-TE)
74	4984	0.468e−11	0.170e−11	315	170
324	17664	0.501e−11	0.171e−11	455	188
1152	57336	0.620e−11	0.195e−11	468	215
2949	139564	0.937e−10	0.234e−10	845	326

The division into more subdomains affects slightly the approximation of the electric field although the relative order of the error in other areas of the domain is not altered.

In the following, hexahedra as shape to discretize the FEM domain is checked. In Table 5.6, different meshes are applied to a cube of dimension 1 m×1 m×1 m using ParMETIS to divide the problem into 4 subdomains, order $p = 3$ for the basis functions and monomial $\mathbf{E}_{\text{MMS}} = (x^2y^2z^2, 0, 0)$ which belongs to the space of basis functions for $p = 3$. Same effect related to numerical error reported in former sections is obtained. The iterations needed to achieve desired convergence for the global surface problem are also included.



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 5.24: Results of MMS test with a monomial belonging to the space of functions of tetrahedra of order $p = 4$ with a cube as domain divided into 4 subdomains through ParMETIS

A graphical representation of the last solution obtained in Table 5.6 is included in Figure 5.25. The same conclusions as in previous section can be extracted: although the division into subdomains can be distinguished in the error, approximation of the solution does not have any discontinuity because of that division.

To end with the test of the three available different shapes, hierarchical vector basis functions of order $p = 3$ applied to triangular prisms are used in Table 5.7 with monomial $\mathbf{E}_{\text{MMS}} = (0, 0, x^3 z^2)$ and same cubic domain as for hexahedra generating 4 subdomains through ParMETIS. The numerical error increases when the mesh is smaller, attributed to the same explanation commented in previous section. In Figure 5.26, the same set of graphical results as in other sections are included, where the divisions in subdomains is detected but the approximation of the solution is not affected.

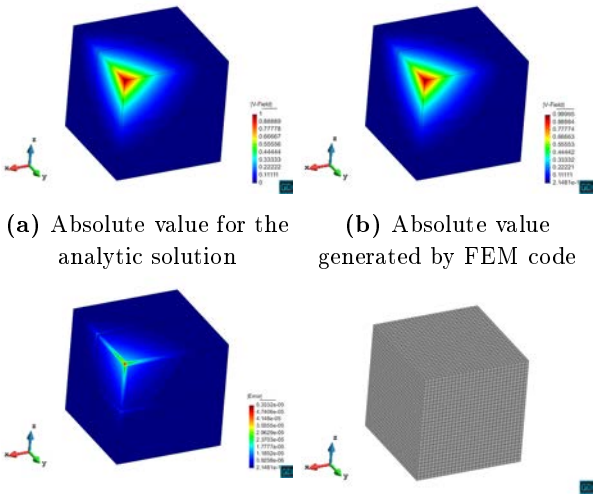
Now, a higher number of domains obtained through ParMETIS is tested. Tetrahedra are chosen to discretize the electromagnetic problem since non-planar interfaces are more demanding from the point of view

Table 5.6: Relative errors for hierarchical hexahedra ($p = 3$) using 4 domains through METIS

Elements	*Unknowns*	*Direct solver*		Iterative solver	
		ς	ς_{curl}	(FOTC)	(SOTC-TE)
64	8040	0.157e−12	0.565e−13	171	142
512	52320	0.846e−12	0.103e−12	235	206
1000	97800	0.775e−11	0.541e−11	261	228
3375	311859	0.606e−10	0.668e−10	354	314

Table 5.7: Relative errors for hierarchical triangular prisms ($p = 3$) using 4 subdomains through ParMETIS

Elements	*Unknowns*	*Direct solver*		Iterative solver	
		ς	ς_{curl}	(FOTC)	(SOTC-TE)
16	1500	0.594e−12	0.139e−12	179	113
260	16257	0.998e−12	0.190e−12	320	195
2080	111393	0.775e−11	0.108e−11	372	189
4272	220983	0.300e−10	0.339e−10	552	292



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

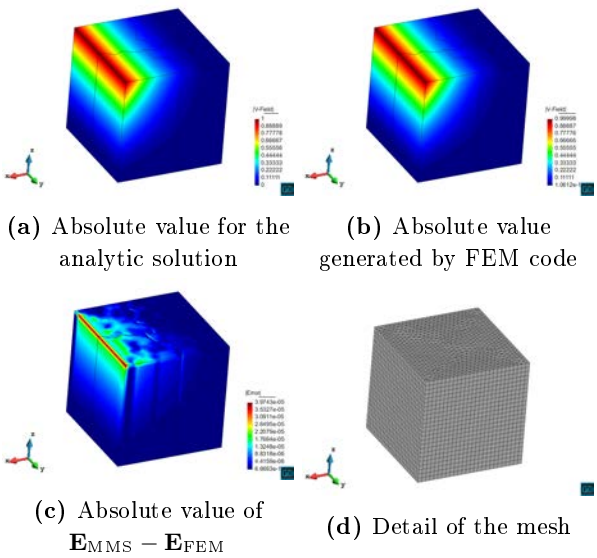
(d) Detail of the mesh

Figure 5.25: Results of MMS test with a monomial belonging to the space of functions of hexahedra of order $p = 3$ with a cube as domain to solve and using METIS to create the division into subdomains

of implementation. A cube with dimensions of $10\,\text{m} \times 10\,\text{m} \times 10\,\text{m}$ decomposed into 24 subdomains is simulated, using a smooth exponential function as excitation and systematic family of basis functions (order $p = 2$) to approximate the electric field. The desired accuracy in the iterative solver is achieved in 554 and 181 iterations with FOTC and SOTC-TE respectively, while an error of 0.4497e−2 is achieved, which is the same error as the one obtained without DDM. In Figure 5.27 the solution is shown where no red areas on the interface between subdomains are detected.

Finally, non-uniform p refinement with DDM is tested next. The cube shown in Figure 5.28 is simulated. For illustration purposes, tetrahedra of order $p = 2$ have been used, and entities (edges or faces) with their x-coordinate in the range $[0.2-0.8]$ are discretized with $p = 3$. The problem has been divided into 4 conformal and non-planar subdomains through METIS, and results are shown in Figure 5.28.

Once several results regarding domains discretized with the same shape have been introduced, the effect of using different shapes for each



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

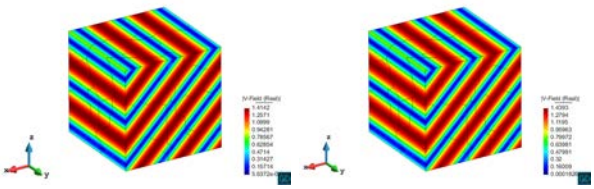
(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 5.26: Results of MMS test with a monomial belonging to the space of functions of triangular prism of order $p = 3$ with a cube as domain to solve and using METIS to create four subdomains

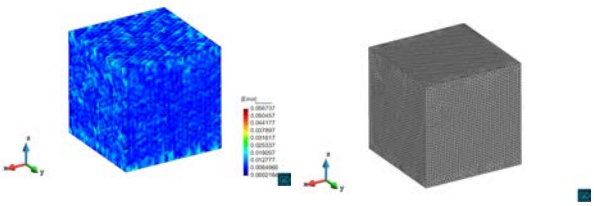
subdomain with a matching mesh on the interface between subdomains is tested. Thus, two different combinations can be checked: tetrahedra with triangular prisms using triangular faces on the interface, and hexahedra with triangular prism using rectangular faces on the interface.

First, a $1\,\text{m}^3$-cube is discretized with a mix of tetrahedra and triangular prisms and a smooth exponential function is approximated with hierarchical vector basis functions of order $p = 2$. as shown in Figure 5.29. With that problem, 233 iterations are needed to solve the surface problem with FOTC while, with SOTC, the solution is obtained in 136 iterations with an error of 0.1478e−3 for both TC. Regarding the number of iterations, with SOTC, it is reduced as anticipated in Section 5.2, while to detect if some loss of accuracy is generated due to domain decomposition, a simulation with the same problem discretized with tetrahedra of the same size as in Figure 5.29 is performed, obtaining an error of 0.1482e−3. Hence, no loss of accuracy is obtained when applying DDM between different shapes of finite elements using a conformal interface.



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code



(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

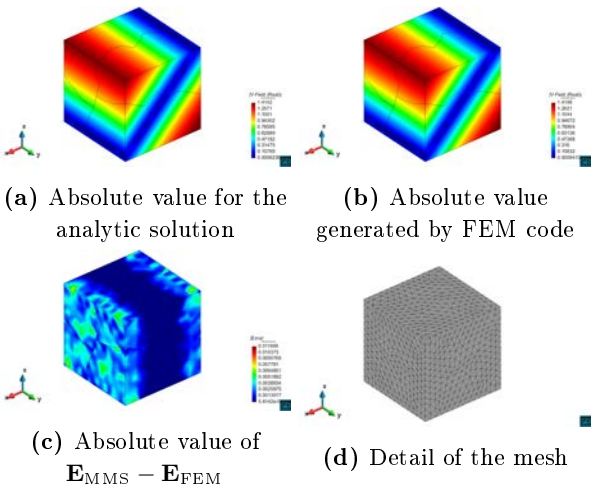
(d) Detail of the mesh

Figure 5.27: Results of MMS test of a 10 m-cube with a smooth exponential function approximated by systematic basis functions and a division into 24 subdomains obtained through ParMETIS

In the following, the same domain is used with a combination of triangular prisms and hexahedra as shown in Figure 5.30 using hierarchical vector basis funcitons of order $p = 4$, and obtaining the solution in 243 and 170 iterations with FOTC and SOTC-TE respectively, both with an error of 0.2854e−7. When solving the same problem with prisms of the same size and order the error obtained is 0.2694e−7, so no loss of accuracy is obtained with matching meshes even although the the interface is non-conformal.

So far, although the formulation and implementation shown in this dissertation supports non-matching meshes on the interface, the effect has not been studied yet. In the following, three non-matching meshes are tested: triangular with triangular faces, rectangular with rectangular faces and, finally, triangular with rectangular faces.

First, the domain shown in Figure 5.31 is divided into two subdomains meshed independently with tetrahedra and generating a non-matching mesh on the interface. 151 iterations for SOTC-TE and 2739 for FOTC are needed to solve the problem, showing the results in Figure 5.31. No abnormalities in the distribution of the error are detected



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

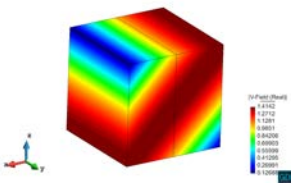
(d) Detail of the mesh

Figure 5.28: Results of MMS test with a smooth function in a mesh with conformal division with different p in the same mesh

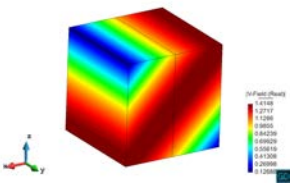
due to the non-conformality between subdomains.

Now, let's consider non-conformality between rectangular faces. In Figure 5.32, hierarchical vector basis functions are used with order $p = 2$, and a hotspost is detected in the non-conformal interface between subdomains, due to the difference in the approximation between hexahedra and prisms but the final solution is not affected. The number of iterations needed in this case is 117 for SOTC and 3458 for FOTC.

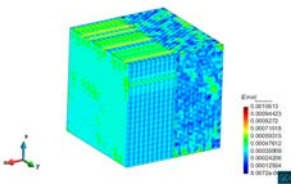
Finally, the interface between triangular and rectangular faces is tested. First, the domain shown in Figure 5.33 is divided into two subdomains, one meshed with tetrahedra and the other with triangular prisms in the way that on the interface there is a interaction between triangular and rectangular faces. In the results, no distinctive performance is obtained with respect to other combinations even when different orders have been used for the different subdomains ($p = 2$ for the case of tetrahedra and $p = 3$ for triangular prisms). Regarding the number of iterations, 224 iterations for SOTC and 4148 for FOTC are needed to obtain the solution. No additional comments to the already explained for Figure 5.32 can be extracted, so the effect of non-conformality between meshes is very reduced when using smooth exponential functions.



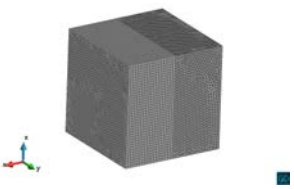
(a) Absolute value for the analytic solution



(b) Absolute value generated by FEM code

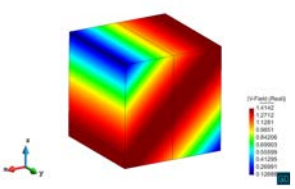


(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

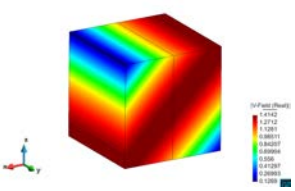


(d) Detail of the mesh

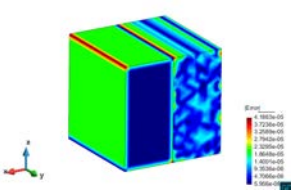
Figure 5.29: Results of MMS test with a smooth function in a mesh with conformal division with tetrahedra and triangular prisms



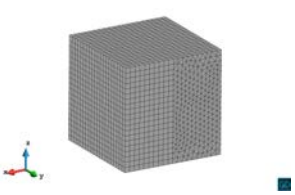
(a) Absolute value for the analytic solution



(b) Absolute value generated by FEM code

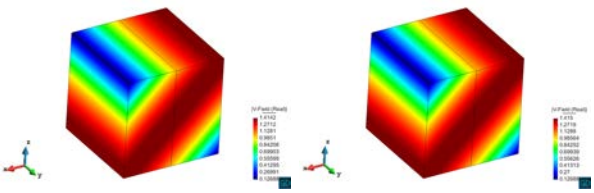


(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$



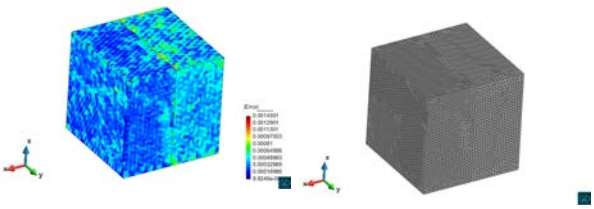
(d) Detail of the mesh

Figure 5.30: Results of MMS test with a smooth function in a mesh with conformal division with hexahedra and triangular prisms



(a) Absolute value for the analytic solution

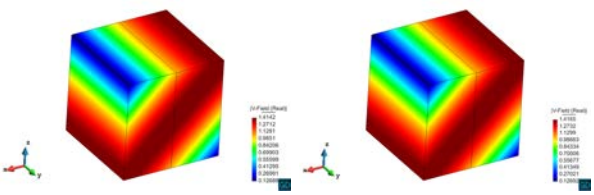
(b) Absolute value generated by FEM code



(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

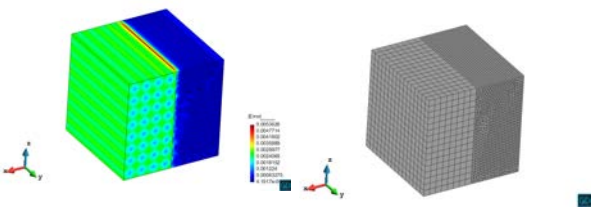
(d) Detail of the mesh

Figure 5.31: Results of MMS test with a smooth function in a mesh with non-conformal division with tetrahedra



(a) Absolute value for the analytic solution

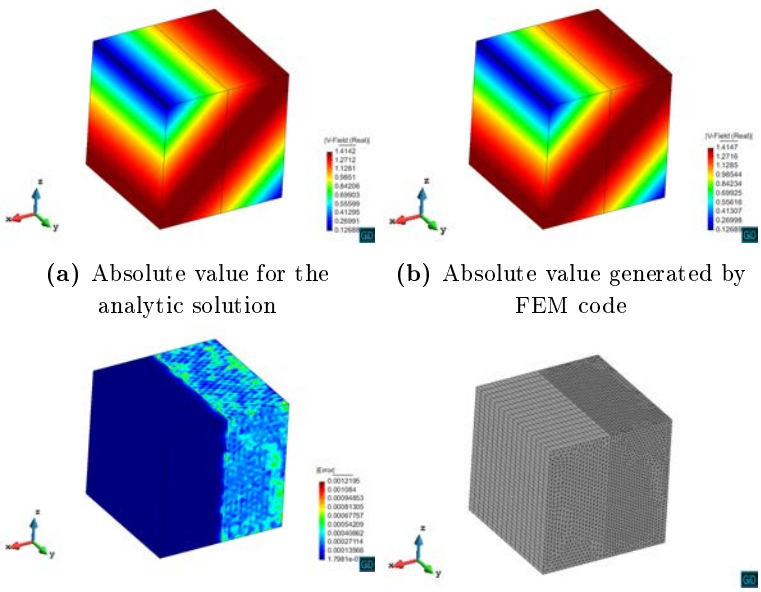
(b) Absolute value generated by FEM code



(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 5.32: Results of MMS test with a smooth function in a mesh with non-conformal division with triangular prisms and hexahedra



(a) Absolute value for the analytic solution

(b) Absolute value generated by FEM code

(c) Absolute value of $\mathbf{E}_{\text{MMS}} - \mathbf{E}_{\text{FEM}}$

(d) Detail of the mesh

Figure 5.33: Results of MMS test with a smooth function in a mesh with non-conformal division with triangular prisms ($p = 3$) and tetrahedra ($p = 2$)

5.4 Real structures

Real structures are used here to validate DDM introduced in this work. Results obtained with DDM are compared against the results obtained without using DDM. Note that, generally, no direct comparison with any measurements or analytical solution is included for the sake of brevity since the code without DDM has already been tested in a number of papers showing a good agreement with other commercial softwares and measurements, [148].

In terms of the propagation of the domains, problems simulated in this section can be classified into one-, two- and three-dimensional problems. Here, the dimension is related to the direction in which the division into subdomains is performed: e.g., a waveguide is a one-dimensional problem since the division into subdomains is introduced along the longitudinal face whereas a two-dimensional antenna array is a two-dimensional problem if each radiating element is considered as a different subdomain. A representative set of different electromagnetic problems is included in this section, covering the following topics: propagation in waveguides, interface between subdomains with different materials, frequency filters with dielectric resonators, computation of far-field directivity and computation of monostatic RCS.

5.4.1 One-dimensional problems

Each one-dimensional problem reported here is based on rectangular waveguides. Despite its simplicity, this propagation problem is one of the most demanding from the point of view of DDM if the division into subdomains is introduced transversal to the longitudinal direction of the waveguide as shown in Figure 5.34 since the convergence of the DDM problem strongly depends on the approximation of propagating and evanescent modes. For each subdomain, all the modes are excited with the introduction of TC and this might be a problem specially for non-conformal interfaces. In this section, PETSc iterative solver using GCR method with a relative tolerance of 1e−6 and a maximum number of iterations set to 5999 has been used, and SOTC-TE are used since a better convergence is achieved in all cases.

First, a conformal division with ParMETIS is introduced in an X-band rectangular WR-90 waveguide with length of 1λ for a working frequency of 7.5 GHz. Systematic family of basis functions particularized for triangular prisms is used and results obtained are included in

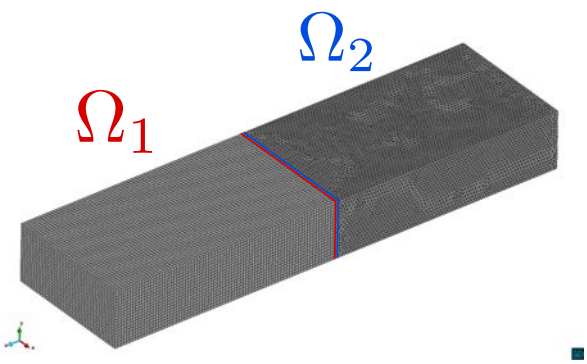


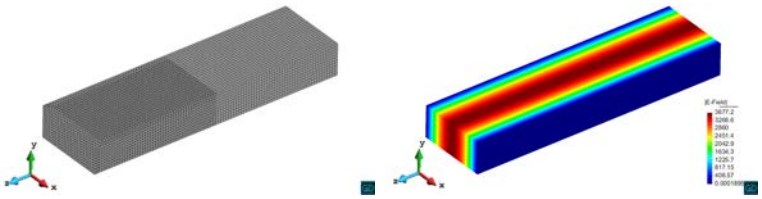
Figure 5.34: Example of division in a WR-90 waveguide

Table 5.8: Results of 1λ-waveguide using ParMETIS

#dom.	*#unkn.*	*#surf. unkn.*	*# iter.*	*Overhead*	$\lvert s_{21} \rvert$
1	81436	—	—	—	0.999999
2	41426	1888	415	1.7 %	0.999999
3	28257	4448	435	4.1 %	0.999999
5	17658	9136	658	8.4 %	0.999999

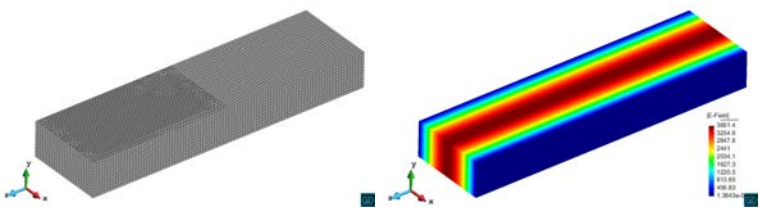
Table 5.8, where # unkn. stands for the mean number of unknowns for each subdomain in the case of DDM, and #dom. is the number of subdomains introduced by ParMETIS. First, it is observed that the accuracy of the results in terms of transmission parameter $|s_{21}|$ is not affected when matching interfaces are constructed, being limited by the resolution of the iterative solver. Overhead as computed in (5.4) is increased with the number of subdomains as expected, while the number of iterations is also higher. This may be expected since the number of unknowns of the surface problem is higher for the same propagation problem. Finally, the number of surface unknowns is increased with the number of subdomains. This increase is linear if planar interfaces are generated: when using ParMETIS, the increase is approximately linear.

Once the impact of using ParMETIS to generate the division into subdomains is evaluated, let's consider a last case of matching meshes on the interface: hybrid meshes in the way of connecting different shapes.



(a) Detail of the mesh **(b)** Absolute value of **E**

Figure 5.35: Results of waveguide with $l = 1\lambda$ and with hybrid mesh and matching mesh with triangular faces

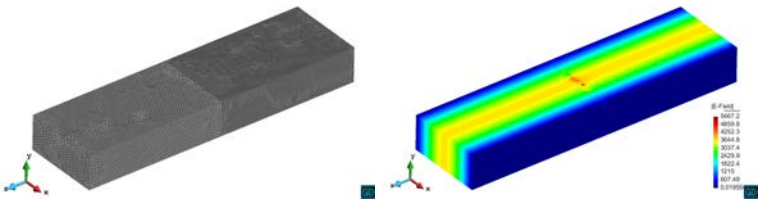


(a) Detail of the mesh **(b)** Absolute value of **E**

Figure 5.36: Results of waveguide with length $l = 1\lambda$ and with hybrid mesh and matching mesh with rectangular faces

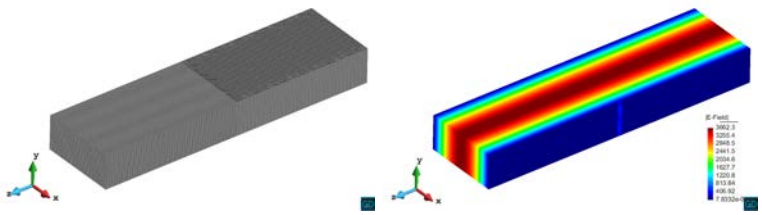
First, tetrahedra and triangular prisms are connected in the same waveguide of length $l = 1\lambda$ used before as shown in Figure 5.35 and divided into two subdomains, each one of them corresponding to a section of the waveguide of length $l = 0.5\lambda$. Numerical details are included in Table 5.9, while the representation of the field is present in Figure 5.35. Note that no loss of accuracy is detected and convergence is good as expected. Next, triangular prisms and hexahedra are connected using the same decomposition into subdomains as shown in Figure 5.36, drawing similar conclusions from the results shown in Table 5.9.

Now, let's consider non-matching meshes on the interface. First, the same division into two subdomains is meshed with unstructured tetrahedra obtaining results included in Table 5.9. A non-negligible loss of accuracy is obtained due to this non-conformality, so a finer mesh is tested, obtaining similar results but with a reduced number of iterations. Results with the finer mesh are shown in Figure 5.37, obtaining a discontinuity on the interface which affects strongly the propagation



(a) Detail of the mesh **(b)** Absolute value of **E**

Figure 5.37: Results of waveguide with length $l = 1\lambda$ and meshed with unstructured tetrahedra and non-matching mesh on the interface

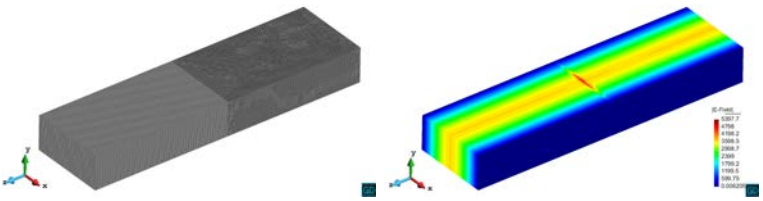


(a) Detail of the mesh **(b)** Absolute value of **E**

Figure 5.38: Results of waveguide with length $l = 1\lambda$ and meshed with triangular prisms and hexahedra with a non-matching mesh on the interface

close to that interface but slightly to the whole solution (hence the loss of accuracy). In the case of non-matching meshes between rectangular faces, hexahedra and triangular prisms are used to discretize each subdomain as included in Table 5.9, where a coarse and a fine mesh are tested; and results for the fine mesh are shown in Figure 5.38. The loss of accuracy is not as remarkable, as observed in Figure 5.37 where the fine mesh is represented. However, the number of iterations needed to achieve relative error is higher than for the case of triangular faces.

Last case considered in Table 5.9 is the interaction between triangular and rectangular faces on the interfaces. A coarse and a fine mesh are used, showing best accuracy in the last case with a higher number of iterations. Results and detail of the mesh are shown in Figure 5.39, where the same performance obtained with non-matching meshes with triangular faces are obtained. The most important conclusions that can be drawn from this thorough set of propagation results are that non-



(a) Detail of the mesh **(b)** Absolute value of **E**

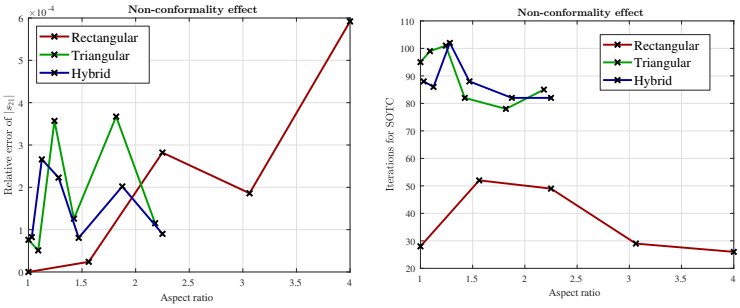
Figure 5.39: Results of waveguide with length $l = 1\lambda$ and meshed with tetrahedra and hexahedra

Table 5.9: Results of 1λ-waveguide with a user-driven division into 2 subdomains

Case	*#unkn. 1*	*#unkn. 2*	*#iter.*	$\lvert s_{21} \rvert$
Hybrid match pri-tet	53400	71544	354	0.999997
Hybrid match hex-pri	46200	41426	717	0.999999
Non-match tet coarse	78640	60038	732	0.994761
Non-match tet fine	469484	186688	266	0.998458
Non-match hex-pri coarse	46200	64902	394	0.999999
Non-match hex-pri fine	197672	204504	3018	0.999984
Non-match tet-pri coarse	78640	41426	526	0.994267
Non-match tet-hex fine	469484	204504	1710	0.999288

conformal interfaces introduce a non-negligible but acceptable source of error, and that the interface between elements is not a source of error by itself since with matching meshes no loss of accuracy is detected. Specifically, from Table 5.9, as expected by FEM theory, connection between different finite element shapes does not introduce an additional source of error, for both matching and non-matching interfaces. This provides flexibility to the code presented in this dissertation since different elements can be assigned to each subdomain depending on the geometry of the particular subdomain without adding any additional error. Finally, from the same set of results, it is worth noting that more accurate results are not obtained with a finer (and uniform) refinement on the non-matching interface.

A further study of the non-conformality follows to assess the effect of



(a) Relative error **(b)** Number of iterations

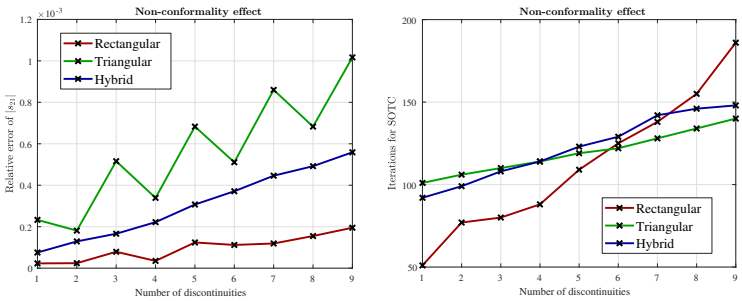
Figure 5.40: Effect of non-conformality in a WR-90 waveguide of one wavelength

the non-matching interface in the error for a propagation problem. To test the difference between the size of the elements on the interface, the same problem is meshed with a given size for the volumetric mesh but changing the number of elements on the interface for the domain Ω_2 to modify the aspect ratio (as a measurement of the difference in the size of elements). The error is computed here as

$$\varsigma_{\text{prop}} = \frac{|s_{21,\text{DDM}} - s_{21,\text{no DDM}}|}{|s_{21,\text{no DDM}}|}, \tag{5.7}$$

and a direct solver is used for the surface problem to measure the relative error in order not to mask the error with the accuracy of the iterative solver as shown in Figure 5.40a. It is observed that the error is quite flat (note the scale on Y axis), and no additional error is detected from the interface between triangular and rectangular faces. Regarding the number of iterations when an iterative solver is used, as included in Figure 5.40b, no special correlation is obtained from the aspect ratio but better results are appreciated for rectangular faces.

Secondly, the original WR-90 waveguide with length $l = 1\lambda$ becomes in a waveguide with length $l = 0.5N\lambda$, N being the number of subdomains introduced. Thus, the original problem sets $N = 2$ and wave propagation only goes trough one discontinuity. Now, the effect of going trough $N - 1$ discontinuities is to be studied, so an aspect ratio of 1.5625 is set on the interface and same results are obtained in Figure 5.41 up to nine discontinuities (generated with ten subdomains). Interesting



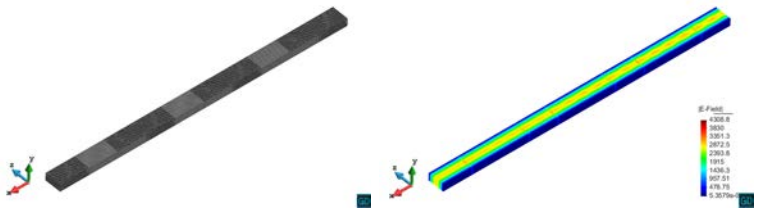
(a) Relative error **(b)** Number of iterations

Figure 5.41: Effect of non-conformality in a WR-90 waveguide with variable wavelength

conclusions can be drawn from this set of results: *i)* a clear trend can be observed in the error which is more pronounced for triangular non-matching interfaces although the dynamic range of the Y axis is still quite reduced; and *ii)* a quasi-linear growth of the number of iterations is observed.

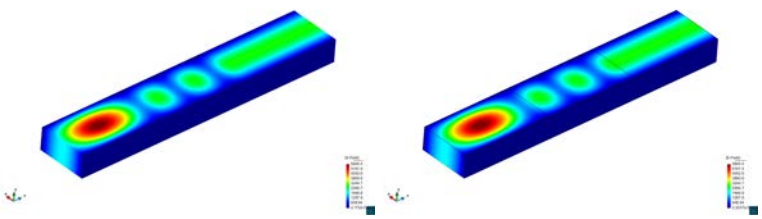
Finally, a waveguide with length $l = 10\lambda$ with the same working frequency is simulated, where each section with length $l = 1\lambda$ corresponds to one subdomain discretized with a different shape: tetrahedra-hexahedra-prism-tetrahedra-hexahedra-prism-tetrahedra-hexahedra-prism-tetrahedra. Matching meshes on the interface are used for interfaces between triangular prisms and hexahedra, and the solution is obtained in 786 iterations and shown in Figure 5.42, obtaining a transmission parameter of $|s_{21}| = 0.995835$. It can be seen that a loss of accuracy is present due to the non-matching interfaces between triangular and rectangular faces as expected from above results, and that the whole solution of the problem is slightly affected by this mismatch.

To test different materials within the same waveguide, let's assume three waveguide sections with length $l = 1\lambda$, where the first and third sections are filled with air while the second one is filled with a dielectric material of $\varepsilon_r = 2$. Each section corresponds to one subdomain, so a problem with three subdomains is considered. The fact that the interface between subdomains corresponds to the interface between materials might be problematic since propagating and evanescent modes have to



(a) Detail of the mesh **(b)** Absolute value of **E**

Figure 5.42: Results of $10-\lambda$ waveguide meshed with different shapes

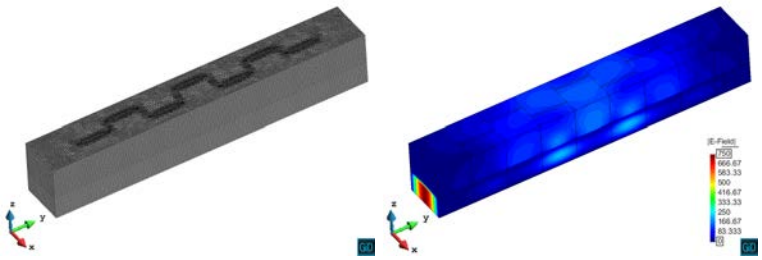


(a) Without DDM **(b)** With DDM

Figure 5.43: Results of a waveguide filled with different materials

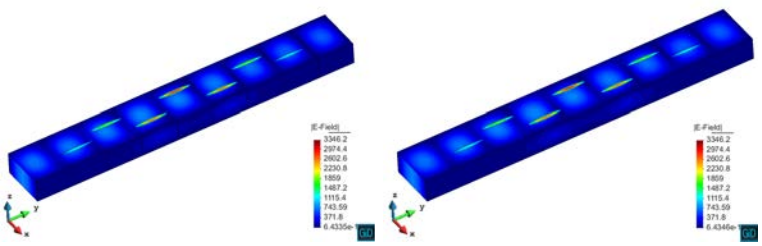
be perfectly approximated. The working frequency is set to $f = 7.5\,\mathrm{GHz}$ which is within the monomode bandwidth of each section, and a user-driven conformal division is introduced in the problem, which is fully discretized with triangular prisms. Results obtained are shown in Figure 5.43, where it can be seen that no loss of accuracy is present when using DDM, and the convergence of the iterative solver is achieved in 631 iterations.

Next case to be analyzed is a slotted waveguide array (SWA) array of 7 elements, [149]. This structure is a good match for being discretized with triangular prisms due to the geometry of the problem as shown in Figure 5.44 due to the fact that the slots are designed to be elliptical instead of rectangular. The working frequency of this array is $3.4045\,\mathrm{GHz}$ and the physical length of the structure is 4.5λ. Nine user-driven subdomains are generated, creating nine 0.5λ sections with a matching mesh on the interface between subdomains. The systematic family of basis functions for triangular prisms is used. It has to be noted that a box for mesh truncation is added for computing far-field results (e.g., directiv-



(a) Absolute value of **E** **(b)** Detail of the mesh

Figure 5.44: Near-field in the whole problem for elliptical SWA array



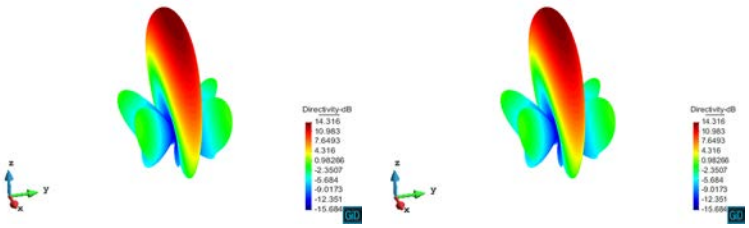
(a) With DDM **(b)** Without DDM

Figure 5.45: Near-field in the waveguide for elliptical SWA array with and without DDM

ity). This box is shown in Figure 5.44 to show the mesh and to detect any anomalies in the division between subdomains.

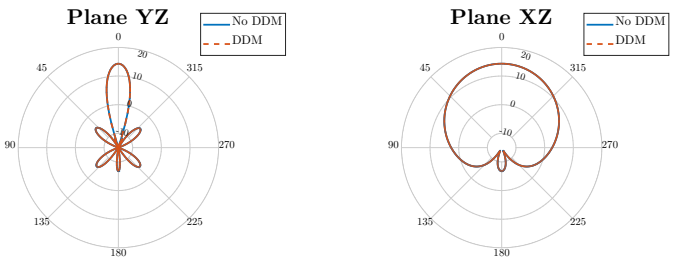
Near-field results with and without DDM are included in Figure 5.45, showing no differences. Finally, three-dimensional and two-dimensional directivity results are included in Figure 5.46 and Figure 5.47, showing again same results. In this case, the accuracy of the DDM is enough not to perceive any difference in the results.

Last one-dimensional problem to be tested is a rectangular cavity filter based on dielectric resonators, [150]. In this problem, four dielectric resonators of $\varepsilon_r = 30$ are embedded in rectangular cavities to generate resonances for filtering a given range of frequencies in the X-band. Details of the model and exact geometry can be found in [150]. Six domains are generated through the division of the slots which communicates the four cavities and the two waveports. The division has to be introduced



(a) With DDM **(b)** Without DDM

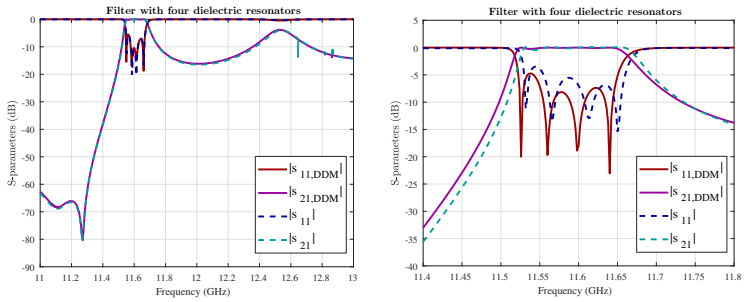
Figure 5.46: 3-D directivity (in dB) for the elliptical SWA array with and without DDM



(a) With DDM **(b)** Without DDM

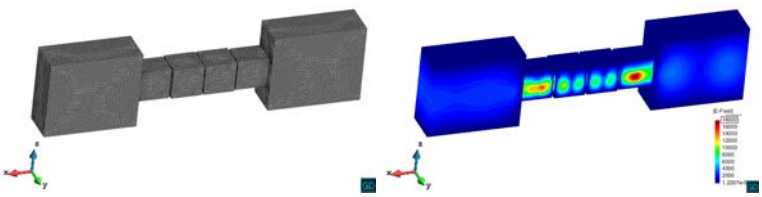
Figure 5.47: 2-D directivity (in dB) in two different planes for the elliptical SWA array with and without DDM

in the middle of the slots for having the same modes on both sides; other than that, convergence problems may appear. The whole problem when not using DDM and each subdomain are meshed with unstructured tetrahedra. Non-matching meshes are used on the interfaces but elements employed are of comparable size in an attempt to minimize the accuracy errors due to non-conformalities. In Figure 5.48 a frequency sweep covering the working frequency range is simulated and compared to the results obtained without using DDM using comparable meshes in terms of the element size. A good agreement is reported, specially taking into account the sensitivity of the results with respect to the mesh due to the high permittivity of the dielectric resonators. Next, a finer mesh is generated to simulate the same problem, and detail of the working range of frequencies is included in the same figure. In this case, the mesh with



(a) X-band frequency sweep **(b)** Finer mesh

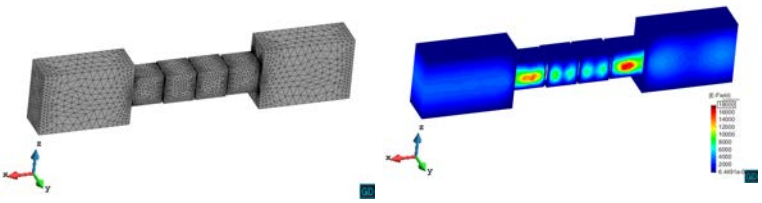
Figure 5.48: Frequency sweep between 11 and 13 GHz of a four dielectric resonator filter with different meshes



(a) Detail of the mesh without DDM **(b)** Absolute value of the electric field obtained without DDM

Figure 5.49: Results in the first resonance for a four dielectric filter without DDM

DDM is more detailed in the rectangular cavities (with the same details as the case without DDM) whereas the section between waveports and first and last slot is meshed more coarsely. A number of 236690 unknowns is obtained for DDM case, whereas to obtain results with similar detail with uniform mesh 412428 unknowns are needed as shown in the meshes included in Figs. 5.49 and 5.50. A good agreement is reported as well, with deeper resonances for the DDM case. Finally, near-field is obtained for both meshes for the frequency of the first resonance for each mesh (11530 MHz for DDM and 11536 MHz when not using DDM), showing detail of the mesh employed in Figs. 5.49 and 5.50. Note that similar results are obtained for both meshes.



(a) Detail of the mesh with DDM

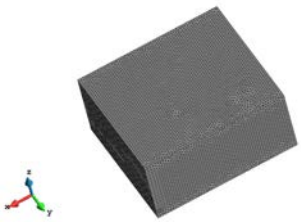
(b) Absolute value of the electric field obtained with DDM

Figure 5.50: Results in the first resonance for a four dielectric filter with DDM

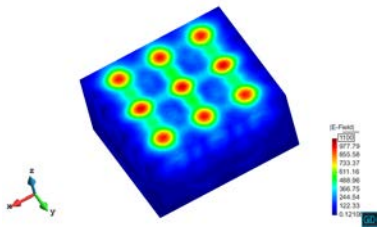
5.4.2 Two-dimensional problem

A two-dimensional array of circular horns is simulated in the following to observe accuracy for this distribution of subdomains. Each element of the array is configured as one subdomain and an array of nine elements is simulated first. The working frequency of the problem is set to 10 GHz, and a uniform excitation is used to generate the solution. Near-field results are shown in Figure 5.51, where a very good agreement between the results is obtained. No division into subdomains are appreciated in the figure. The solution is obtained in 62 iterations while the total number of unknowns without DDM is 1360188 unknowns, and a similar number is obtained with DDM, 1398421 unknowns. Finally, the size of the surface problem is 60144 unknowns.

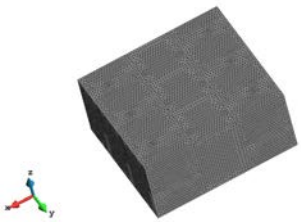
A bigger problem is simulated in the following, increasing the number of elements in the array to sixteen. The same setup is used, obtaining the results shown in Figure 5.52, where no division into subdomains is observed. Moreover, 3-D directivity results are included in Figure 5.53 and cuts in planes XY and XZ are shown in the same figure, where a total agreement between results is obtained. Specific details of the simulation are included in the following: the total number of unknowns is 2261472 and 2368032 without and with DDM respectively. The number of iterations obtained is 73 to achieve the solution shown in Figure 5.52. Note that no special problems arise when using a two-dimensional distribution of the results.



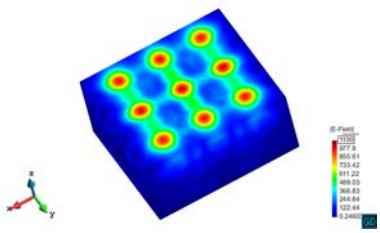
(a) Detail of the mesh without DDM



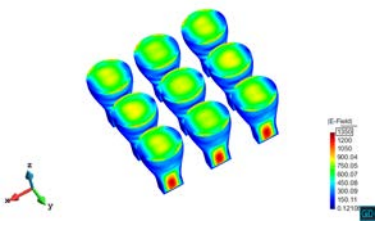
(b) Absolute value of the electric field obtained without DDM and with truncation boundary box



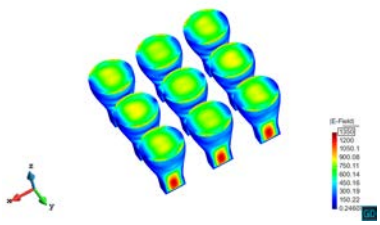
(c) Detail of the mesh with DDM



(d) Absolute value of the electric field obtained with DDM and with truncation boundary box

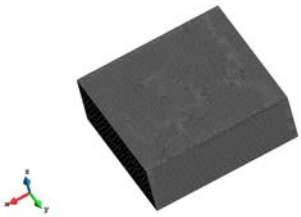


(e) Absolute value of the electric field obtained without DDM

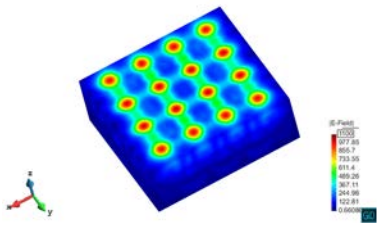


(f) Absolute value of the electric field obtained with DDM

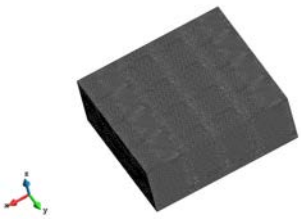
Figure 5.51: Near-field results for a 3x3 array of circular horns



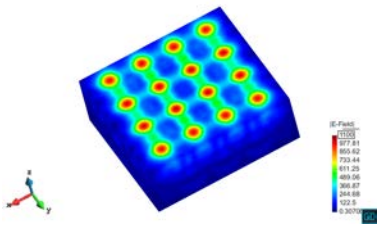
(a) Detail of the mesh without DDM



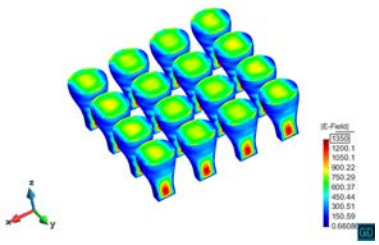
(b) Absolute value of the electric field obtained without DDM and with truncation boundary box



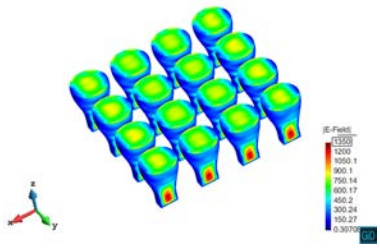
(c) Detail of the mesh with DDM



(d) Absolute value of the electric field obtained with DDM and with truncation boundary box



(e) Absolute value of the electric field obtained without DDM

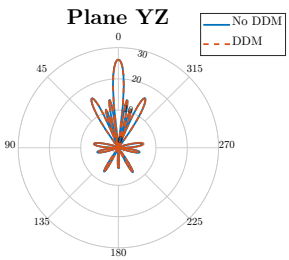


(f) Absolute value of the electric field obtained with DDM

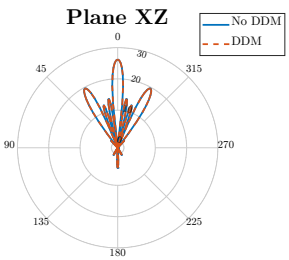
Figure 5.52: Near-field results for a 4x4 array of circular horns

(a) 3-D directivity results without DDM

(b) 3-D directivity results with DDM



(c) 2-D directivity YZ plane with and without DDM



(d) 2-D directivity XZ plane with and without DDM

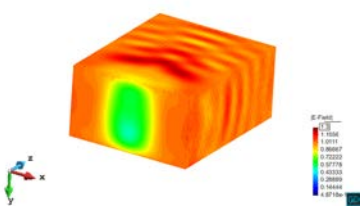
Figure 5.53: Far-field directivity results for a 4x4 array of circular horns

5.4.3 Three-dimensional problem

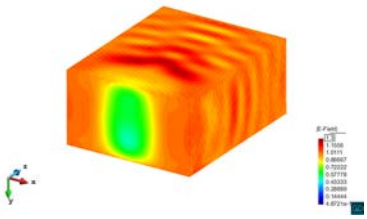
Finally, computation of a RCS over a F117 plane is obtained. The working frequency is set to 50 MHz and different number of domains are introduced with ParMETIS. Note that in this case, the surfaces are not fully straight and some problems can arise on the non-planar interfaces: from the point of view of the implementation, a conformal strategy of detection of the points in the different elements has been introduced, selecting the elements which evaluate the shared points with the vertices of the face and not detecting if a point is geometrically on the face (as it is the case with non-conformal interfaces). Near-field results are shown in Figure 5.54, where five subdomains are used for DDM results. Different cuts of the 3D RCS results are shown in Figure 5.55, where a good agreement is obtained in the results. Specific details of the simulations are included in Table 5.10 where the number of unknowns is the mean number by each subdomain when DDM is used. Note that the overhead is increased with the number of subdomains and that the growth of the number of surface unknowns is unpredictable since the number of interfaces between subdomains is unknown *a priori*, whereas for user-driven domain decomposition this number can be predicted.

Table 5.10: Results of F117 airplane using ParMETIS

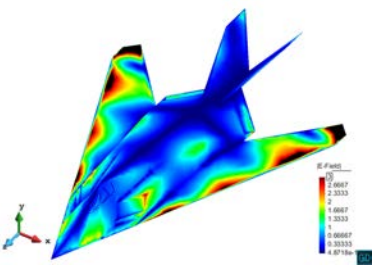
#dom.	*#unkn.*	*#surf. unknowns*	*# iter.*	*Overhead*
1	1011464	—	—	—
2	513346	24304	50	1.5 %
3	348951	47124	62	3.5 %
5	212641	68790	80	5.1 %
10	110208	120298	86	8.9 %
20	56224	149596	109	11.1 %



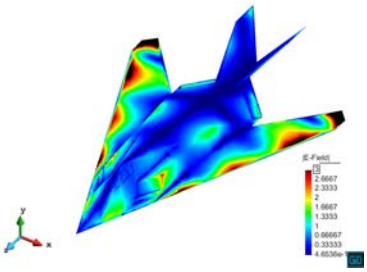
(a) Absolute value of the electric field obtained without DDM and with truncation boundary box



(b) Absolute value of the electric field obtained with DDM and with truncation boundary box

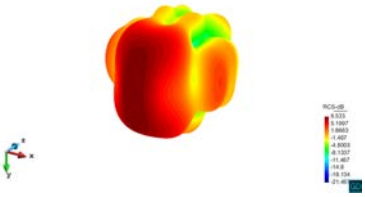


(c) Absolute value of the electric field obtained without DDM



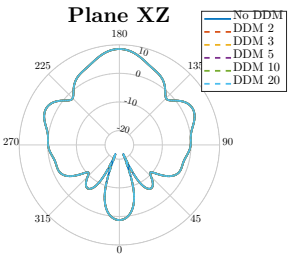
(d) Absolute value of the electric field obtained with DDM

Figure 5.54: Near-field results for F117 airplane

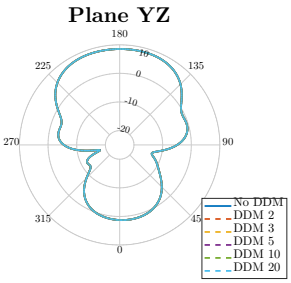


(a) 3D view of RCS with DDM

(b) 3D view of RCS without DDM



(c) XZ plane of RCS results



(d) YZ plane of RCS results

Figure 5.55: Far-field RCS results for F117 airplane

Table 5.11: Performance results for a two-dimensional antenna array

Case of study	*Time (s)*	*Peak mem.(Mb)*	*Unknowns*
3x3 No DDM	416	5380	1360188
3x3 DDM	463	3371	1398118
4x4 No DDM	1579	12253	2261472
4x4 DDM	1191	5832	2368032

5.4.4 Performance

Since advantages in the performance of DDM codes are thoroughly explained in the literature, (see [108] and references therein), only one case of study is included in the following to show the performance of the implementation detailed in this work. Let us consider the two-dimensional antenna array introduced in Section 5.4.2. First, evolution of the residual used by the iterative solver to achieve convergence is shown in Figure 5.56, where a good performance with the increase in the size of the problem is obtained. Moreover, two key indicators have to be considered: memory and computational time. The same number of nodes and MPI processes (5 nodes and 10 processes, with 4 OpenMP threads) have been used to generate results included in Table 5.11, where the peak memory stands for the mean peak memory in each used node, and the number of iterations are obtained with SOTC-TE. It can be observed that the storage of local meshes leads to a non-negligible memory saving in the case of nine and sixteen elements, while regarding computational time, a better number is obtained with a bigger size of the problem. This performance agrees with the implementations shown in the literature, and better numbers can be obtained if, as in this case, only it is stored one element which is virtually repeated to conform the two-dimensional array.

5.5 Conclusions

A detailed procedure to verify the DDM implementation in the code is presented in the first three sections of this chapter. First, a variation of the MMS used in the Chapter 4 to simulate the communication with the other subdomains is presented, using triangular prisms to show numerical results. These results show that an error close to machine precision is

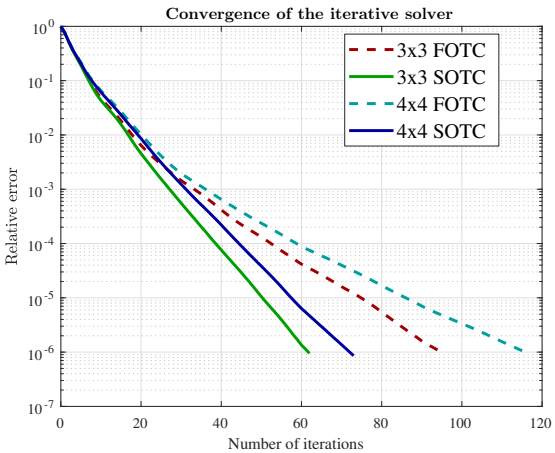


Figure 5.56: Iterative solver convergence for the circular horn antenna array

obtained when using monomials inside of the space of basis functions, and the same accumulation of numerical error is detected. Direct solvers are used to obtain the solution of the surface problem in order not to mask the results with the accuracy of the solver.

In Section 5.2, the eigenspectra for a number of problems is computed to provide an *a priori* estimation of the convergence of the iterative solver when solving the preconditioned global surface problem. A representative set of combinations covering all the finite element shapes and families introduced in Chapter 4 and non-matching interfaces is included, showing no special differences due to the finite element used. SOTC provide a more clustered eigenspectra around the center of the circle than FOTC so convergence of the iterative solver is expected to be better with SOTC. Furthermore, the convergence of the iterative solver is assured in all the results presented since all the eigenvalues are within the circle of convergence.

Then, since only the formulation is tested in Section 5.1, in Section 5.3 the machinery associated to the cross matrices which supports the communication between subdomains is checked through the classic MMS techniques already used in the Chapter 4. Also, iterative solver is introduced to check the convergence of all the problems included in the

section. The monomials inside the space of basis functions together with smooth complex exponential functions are used as manufactured solutions and similar performance as in Chapter 4 is obtained for conformal meshes independently of the finite element used. Non-conformalities show a source of error which slightly affects the approximation of the electric field in the whole problem, so this will be further studied in real problems in Section 5.4.

A detailed study related to real problems is included in Section 5.4 to validate the DDM introduced. Three different sets of results are included depending on the expansion of the number of subdomains, and a representative set of real problems is chosen involving wave propagation problems, different dielectric media, radiation, high-frequency filters and RCS computations. Results have a quite good agreement when conformal interfaces are used, whereas non-conformalities introduce a new source of error which is analyzed quantitatively for a WR-90 waveguide. This analysis shows that the aspect ratio between the elements is not a key factor for this problem, and that the error of $|s_{21}|$ is increased with the number of discontinuities but in the worst case it has a small impact in the error. Also, similar results are obtained with a finer discretization of the mesh on the interface. These results are quite revealing for the interaction with *hp* adaptivity: *i)*, this lack of sensitivity in the aspect ratio means that an independent refinement of each subdomain will not increase the error due to the non-conformal interface; *ii)* the division into subdomains have to be introduced taken into account that the error is linearly increased with the number of discontinuities; *iii)* a uniform refinement on the interface does not decrease the error associated to the non-conformal interface. Finally, some performance results are included and results obtained are as good as expected from the literature.

CHAPTER 6

ADAPTIVITY

Adaptivity, introduced in the last decades in FEM codes, is a technique which improves the approximation in those areas of the domain to be solved where the discretization error is higher. Two versions of adaptivity are found in the literature: h-adaptivity, related to the size of the elements, and p-adaptivity, which involves the order of approximation of basis functions. Adaptivity combined with DDM is a field not thoroughly explored yet in the literature for electromagnetism since the DDM are commonly used for large scale problems, so a first proof of concept is introduced in this section.

In Chapter 5, the approximation of the electric field with non-conformal meshes has been validated. A detailed study of the effect of non-conformalities has been included for wave propagation problems in Section 5.4. Moreover, different orders, even in the same domain, can be used when hierarchical vector basis functions (e.g., from [3]) are employed. For illustration purposes, MMS can be used to introduce a smooth exponential function with frequency $f = 100\,\mathrm{MHz}$ in the problem meshed with tetrahedra shown in Figure 6.1. Three subdomains are introduced and different refinements are applied, as introduced in Table 6.1: uniform h refinement, uniform p refinement, and uniform h refinement for the subdomain in the middle whereas uniform p refinement is applied in the neighbor subdomains. Error defined by (4.31) and the number of unknowns are included in Table 6.2, while the graphical representation of the error for the first and last case is included in Figure 6.1. Note that the last case includes a non-conformal interface where the meshes are quite different in size (and order) between each other. As

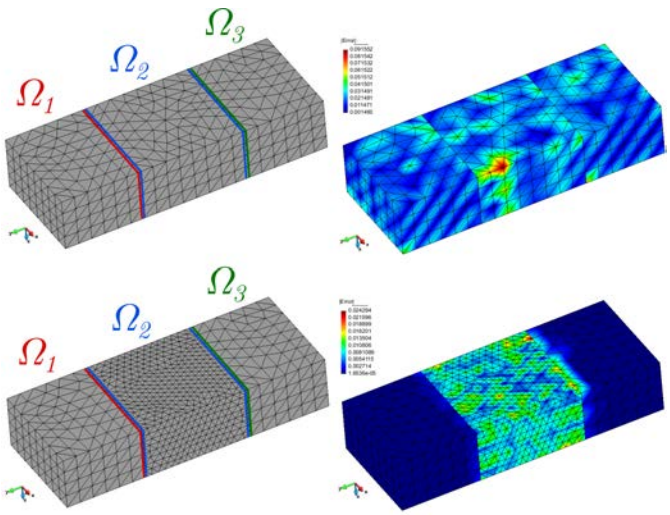


Figure 6.1: Results and meshes used in refinements 1 (above) and 4 (below) as defined in Table 6.1

expected, p refinement is more effective in this problem than h refinement due to the regular geometry of the problem and the smoothness of the function to be discretized. Regarding the last case, the error in h-refined Ω_2 is not affected by the mismatch on the interface between subdomains; however, this refinement increases the error in subdomains Ω_1 and Ω_3 as shown numerically in Table 6.2. Indeed, in Figure 6.1 it can be seen that in the vicinity of the interface the error is increased but not in the interior of subdomains Ω_1 and Ω_3.

Table 6.1: Parameters of the simulations included in Table 6.2

	Parameters of the refinement applied						
	h_1	h_2	h_3	p_1	p_2	p_3	*Unknowns*
1	0.5 m	0.5 m	0.5 m	2	2	2	5848
2	0.25 m	0.25 m	0.25 m	2	2	2	42638
3	0.5 m	0.5 m	0.5 m	4	4	4	33316
4	0.5 m	0.25 m	0.5 m	4	2	4	39866

Table 6.2: Numerical results for a smooth exponential function

	Numerical error from (4.31)			
	ς_1	ς_2	ς_3	ς_{full}
1. Coarse	0.2e−1	0.2e−1	0.1e−1	0.2e−1
2. h ref.	0.4e−2	0.4e−2	0.4e−2	0.4e−2
3. p ref.	0.5e−4	0.7e−4	0.4e−4	0.5e−4
4. hp ref.	0.3e−3	0.4e−2	0.1e−3	0.2e−2

These results show that the code is ready to introduce uniform p and h refinement in each subdomain. Together with the result shown in Fig. 5.28 about non-uniform p refinements in the whole domain, all the building blocks but the adaptivity algorithm have already been introduced. Thus, an adaptivity algorithm particularized for triangular prisms from Chapter 4 is implemented. This shape is chosen since a semi-structured adaptivity is provided, suitable for a number of applications; moreover, as far as the author knows, this adaptivity has not been thoroughly developed in the literature, [129]. However, in terms of formulation, same concerns independently of the element chosen have to be addressed in the application of adaptivity to DDM. In Section 6.1, details about the algorithm itself are provided, and verification through MMS is applied in Section 6.2.

Finally, the final objective of this chapter is to further study the effect of non-uniform adaptivity with non-conformal DDM. In Section 6.3, the effect of DDM is studied first for a WR-90 waveguide and, then, a singularity in the same waveguide is introduced as a proof of concept of the effect of the interface in the adaptivity. Some hp meshes are introduced as well but no special techniques for p refinement have been introduced.

6.1 Algorithm

An adaptivity algorithm can be summarized in four steps:

- Solve the problem.
- Estimate the error in the solution.
- Mark elements to be refined.
- Refine each element.

These four steps are repeated until the convergence criteria are achieved (a maximum number of iteration of a given threshold of the error estimation). In previous chapters, the first step has been explained thoroughly, while for remaining steps many references are available: the focus of this work is not set on proposing disruptive solutions for each one of these steps but applying different alternatives detailed in the literature to the adaptivity algorithm developed here to be merged with DDM formulation. However, since the use of triangular prisms is not standard in FEM, and DDM formulation has to be taken into account in the adaptivity, some contributions have to be introduced in this section.

6.1.1 Estimator

Different choices are available to estimate the error generated by some mesh in the code. A good estimator is a key factor when implementing adaptive strategies since each marking strategy relies on this estimation. Here, a straightforward local estimator has been introduced in the code inspired by [121]. This estimator is computed individually for each element m and can be divided into a *volumetric* residual $\mathcal{R}^{(m)}_{\text{vol},i}$ and a *face* residual $\mathcal{R}^{(m)}_{\text{face},i}$, where superscript $\bullet^{(m)}$ is used here to constrain some magnitude to an element (m). Volumetric residual $\mathcal{R}^{(m)}_{\text{vol},i}$ is defined as

$$\mathcal{R}^{(m)}_{\text{vol},i} = \boldsymbol{\nabla} \times \mu_{ri}^{-1}(\boldsymbol{\nabla} \times \mathbf{E}^{(m)}_{i,\text{FEM}}) - k_0^2 \varepsilon_{ri} \mathbf{E}^{(m)}_{i,\text{FEM}} - \mathbf{O}_i, \tag{6.1}$$

where $\mathbf{E}^{(m)}_{i,\text{FEM}}$ is the solution provided by the code, i.e., $\mathbf{E}^{(m)}_{i,\text{FEM}} = \sum_{k=1}^{N_i} g^{(m)}_{i,k} \mathbf{N}^{(m)}_{i,k}$ with $g^{(m)}_{i,k}$ being the DOFs obtained by the code, $\mathbf{N}^{(m)}_{i,k}$ the basis function associated to that DOF and N_i the number of basis functions. The notation used is the same as defined in (2.20). As usual, the subscript $\bullet_i$ is referred to the subdomain to which the element belongs.

The face residual depends on the boundary condition, if any, applied to the face. With PEC the residual is not defined since Dirichlet BC are imposed to the electric field, whereas a residual is defined when the electric field is not enforced. First, the face residuals associated to

Neumann and Cauchy BC, $\mathcal{R}_{\mathrm{N}}$ and $\mathcal{R}_{\mathrm{C}}$, are defined respectively as

$$\mathcal{R}_{\mathrm{N}}^{(m)} = \hat{\mathbf{n}}_i^{(m)} \times \mu_{ri}^{-1}(\boldsymbol{\nabla} \times \mathbf{E}_{i,\mathrm{FEM}}^{(m)}), \quad \text{on } \Gamma_{i,\mathrm{N}}, \tag{6.2}$$

$$\mathcal{R}_{\mathrm{C}}^{(m)} = \hat{\mathbf{n}}_i \times \mu_{ri}^{-1}(\boldsymbol{\nabla} \times \mathbf{E}_{i,\mathrm{FEM}}^{(m)}) + jk_0\hat{\mathbf{n}}_i^{(m)} \times \hat{\mathbf{n}}_i^{(m)} \times (\boldsymbol{\Psi}_i - \mathbf{E}_{i,\mathrm{FEM}}^{(m)}), \quad \text{on } \Gamma_{i,\mathrm{C}}. \tag{6.3}$$

Then, the face residual between two neighbor elements m and n in a given subdomain i, $\mathcal{R}_{i,\mathrm{neigh}}$, is defined as the error in the tangential continuity of the magnetic field between elements through

$$\mathcal{R}_{i,\mathrm{neigh}}\hat{\mathbf{n}}_i^{(m)} = \hat{\mathbf{n}}_i^{(m)} \times \mu_{ri}^{-1}(\boldsymbol{\nabla} \times \mathbf{E}_{i,\mathrm{FEM}}^{(m)}) + \hat{\mathbf{n}}_i^{(n)} \times \mu_{ri}^{-1}(\boldsymbol{\nabla} \times \mathbf{E}_{i,\mathrm{FEM}}^{(n)}). \tag{6.4}$$

This is motivated by the fact that the electric field is tangentially continuous by FEM assembly of the curl-conforming functions, but magnetic field is derived from the electric field.

Faces which belong to an interface between subdomains i and j are the last case to be considered. That residual $\mathcal{R}_{ij,\mathrm{DDM}}$ is defined as

$$\mathcal{R}_{ij,\mathrm{DDM}}^{(m)} = \pi_\tau(\mathbf{E}_{i,\mathrm{FEM}}^{(m)}) + \pi_\tau^{\times}(\mu_{ri}^{-1}\boldsymbol{\nabla} \times \mathbf{E}_{i,\mathrm{FEM}}^{(m)}) - \pi_\tau(\mathbf{E}_{j,\mathrm{FEM}}^{(m)}) - \pi_\tau^{\times}(\mu_{rj}^{-1}\boldsymbol{\nabla} \times \mathbf{E}_{j,\mathrm{FEM}}^{(m)}). \tag{6.5}$$

Finally, the residual for each element m is introduced through

$$\mathcal{R}^{(m)} = h^{(m)} c_2(\mathcal{R}_{\mathrm{vol},i}^{(m)}, \mathcal{R}_{\mathrm{vol},i}^{(m)}) + \sum_{k}^{n_{\mathrm{faces}}} K_k h_k^{(m)} c_\tau(\mathcal{R}_k^{(m)}, \mathcal{R}_k^{(m)}), \tag{6.6}$$

where K_k is $\dfrac{1}{2}$ if face k is triangular and 1 if face k is rectangular, n_{faces} is the number of faces in the element, and h is the diameter of the entity (volume or face). In case that $\mathcal{R}_{\mathrm{face},i}^{(m)}$ is not defined (i.e., PEC faces), it is set to null. Note that the operator c_2 has been defined in (2.37), while c_τ is defined as

$$c_\tau(\mathbf{W}, \mathbf{W}) = \iint_\Gamma (\mathbf{W} \cdot \mathbf{W})\, d\Gamma. \tag{6.7}$$

6.1.2 Marking strategies

Once the residual is established for each element, different strategies to mark an element to be refined are defined in terms of this local residual. Five different strategies are used in the results included in this chapter:

- Next-step estimator: inspired by [4], it consists of estimating the error which an element would have if it is refined.
- Quantile: as shown in [5], a percentage of the elements, ordered by the residual from highest to lowest computed in the section 6.1.1, are refined.
- Maximum: inspired by [126], elements with a residual higher than the maximum residual in the mesh multiplied by some parameter θ between 0 and 1 are marked to be refined.
- Fixed-energy fraction: elements are ordered by its residual from highest to lowest and the number of elements to be refined conforms some percentage of the energy of the residual in the whole problem —understood as the sum of the square of the residual in each element—. This strategy is shown in [130].
- SER: inspired by [6], the name stands for (Solve-Estimate-Refine). This algorithm is a mix between maximum and fixed-energy fraction algorithms, and it may be understood as a strategy based on Heaviside step function: as in fixed-energy fraction algorithm, marked elements constitute some percentage of the energy of the whole problem. However, this energy is not reached adding elements individually to be refined, but also with a step function which is lower in each step, in a way that elements to be refined are those with a residual higher than the threshold set by the step function. If this step is configured to select only one element in each step, it is equivalent to the fixed-energy fraction strategy explained before.

Unfortunately, due to the semi-structured characteristics of the triangular prism (element under study in this chapter), some of these marking strategies may not be applied directly if problems of hanging nodes are not treated; i.e., if a *conformation* of the mesh is not performed when elements are marked to be refined. A hanging node occurs when there is no continuity between elements: e.g., when an edge of some element ends in the middle of the edge of other element. This situation generates irregular meshes which requires non-standard basis functions to approximate correctly the electric field, [27].

This so-called here process of *conformation*, explained thoroughly in the following section, makes the number of prism elements to be refined

undetermined when they are marked, since the refinement has to be propagated horizontally and vertically. Regarding DDM, these marking strategies may be applied to each subdomain individually or to the whole problem. Independently of this choice, these marking strategies can make the size of the elements on the interface very unbalanced, yielding an undesirable performance as shown in Section 6.2. For this reason, a conformation based on a rule of thumb (ROT) criterion between subdomains has been implemented as detailed in Section 6.2.3. This conformation provides an increased number of elements to be refined which has to be taken into account when applying some of these marking strategies.

With all these considerations, pseudocodes for each strategy follow. The set of elements of the original domain is noted as $\mathcal{P}$, while $\mathcal{P}_{\mathrm{R}}$ stands for the set of elements marked to be refined, with $\mathcal{P}_{\mathrm{R}} \subset \mathcal{P}$. In Pseudocode 6.1, a slight variation of the algorithm proposed in [4] is introduced in case that the estimation is not enough to produce any new element to refine. This estimation is based on the assumption that the residual has an asymptotic behavior of the form $\mathcal{R} = ch^{\lambda}$, h being the characteristic size of the element. Unfortunately, in some cases that estimation did not provide any new element to be refined and, because of that, a variation has been introduced. It has to be noted that the first iteration of this algorithm produces the marking of all the elements since the threshold is set to null.

In Pseudocode 6.2, details about the implementation of the quantile algorithm to mark elements is presented. Since the number of the refined elements is not known beforehand because of the process of conformation of the mesh, the process has to iterate considering less and less elements until the number of elements to be refined are lower than the given percentage (specified through parameter θ) of elements to be refined.

In the case of the maximum algorithm, no special treatment is applied due to the conformation of the mesh. In Pseudocode 6.3 more details about this marking algorithm are provided. Regarding fixed-energy fraction algorithm, a slight modification to include conformation of the mesh has been introduced in the algorithm, as shown in Pseudocode 6.4. The same adjustment is applied to SER algorithm, as included in Pseudocode 6.5

Pseudocode 6.1 Slight variation of [4] in the k iteration of the adaptivity algorithm to produce a set of elements to be refined

Require: $\mathcal{R}_{k-1}^{(m)}$ ▷ Estimator of the last step
Require: conform_refinement($\mathcal{P}_{\mathrm{R}}$) ▷ Process of conformation to avoid hanging nodes
Input: $\mathcal{R}_{k}^{(m)}$ ▷ Evaluation of (6.6)
Input: $\mathcal{P}$ ▷ Set of original elements (of size N)
Output: $\mathcal{P}_{\mathrm{R}}$ ▷ Set of elements marked to be refined (of size N_{R})

1: **procedure** MARK_NEXTSTEP($\mathcal{R}_{k}^{(m)}$,$\mathcal{P}$,$\mathcal{P}_{\mathrm{R}}$)
2: $\theta \leftarrow 1$
3: $i \leftarrow 0$ ▷ Number of iterations of mark_nextstep
4: **if** $k = 1$ **then**
5: $\mathcal{R}_{2}^{(e)} \leftarrow 0 \quad \forall e \in \mathcal{P}$
6: **else**
7: **for all** $e \in \mathcal{P}$ **do** ▷ Go through each element
8: $\mathcal{R}_{k+1}^{(e)} \leftarrow \dfrac{\left(\mathcal{R}_{k}^{(e)}\right)^2}{\mathcal{R}_{k-1}^{(e)}}$
9: **end for**
10: **end if**
11: **while** $N_{\mathrm{R}} = 0$ **do**
12: $\theta \leftarrow \dfrac{\theta}{10^i}$ ▷ Adjustment of ς_{cut}
13: $\varsigma_{\mathrm{cut}} \leftarrow \theta \cdot \max\{\mathcal{R}_{k+1}^{(1)}, \ldots, \mathcal{R}_{k+1}^{(N)}\}$ ▷ Threshold to be marked
14: **for all** $e \in \mathcal{P}$ **do**
15: **if** $\mathcal{R}_{k}^{(e)} \geq \varsigma_{\mathrm{cut}}$ **then**
16: $e \in \mathcal{P}_{\mathrm{R}}$ ▷ Add to the set of marked elements
17: **end if**
18: **end for**
19: $i \leftarrow i + 1$ ▷ Increase the iteration number
20: **end while**
21: $\mathcal{P}_{\mathrm{R}} \leftarrow$ conform_refinement($\mathcal{P}_{\mathrm{R}}$) ▷ Conform marked elements
22: **end procedure**

Pseudocode 6.2 Slight variation of [5], where a given percentage of elements are marked to be refined

Require: conform_refinement($\mathcal{P}_\mathrm{R}$) ▷ Process of conformation to avoid hanging nodes
Require: order_residual($\mathcal{R}_k$) ▷ Order elements from highest to lowest local residual $\mathcal{R}_k^{(m)}$
Input: $\mathcal{R}_k^{(m)}$ ▷ Evaluation of (6.6)
Input: θ ▷ Percentage of elements to be refined
Input: $\mathcal{P}$ ▷ Set of original elements (of size N)
Output: $\mathcal{P}_\mathrm{R}$ ▷ Set of elements marked to be refined (of size N_R)

1: **procedure** MARK_QUANTILE($\mathcal{R}_k^{(m)}$,θ,$\mathcal{P}$,$\mathcal{P}_\mathrm{R}$)
2: $\theta_\mathrm{R} \leftarrow \theta$
3: $\mathcal{P}_\mathrm{ord} \leftarrow$ order_residual($\mathcal{R}_k$) ▷ Get an ordered set of elements
4: **while** $N_\mathrm{R} \geq \theta N$ **do**
5: **for all** $e \leq \theta_\mathrm{R} \cdot N_\mathrm{R}$ **do**
6: $e \in \mathcal{P}_\mathrm{R}$ ▷ Add to the set of marked elements
7: **end for**
8: $\mathcal{P}_\mathrm{R} \leftarrow$ conform_refinement($\mathcal{P}_\mathrm{R}$) ▷ Conform marked elements
9: $\theta_\mathrm{R} \leftarrow \theta - 0.05$ ▷ Try with a smaller set
10: **end while**
11: **end procedure**

Pseudocode 6.3 Marking strategy based on evaluating the residual of each element and compare it to a threshold

Require: conform_refinement($\mathcal{P}_\mathrm{R}$) ▷ Process of conformation to avoid hanging nodes
Input: $\mathcal{R}_k^{(m)}$ ▷ Evaluation of (6.6)
Input: θ ▷ Percentage to set threshold
Input: $\mathcal{P}$ ▷ Set of original elements (of size N)
Output: $\mathcal{P}_\mathrm{R}$ ▷ Set of elements marked to be refined (of size N_R)

```
1: procedure MARK_MAXIMUM(R_k^(m),θ,P,P_R)
2:     ς_cut ← θ·max{R_k^(1), ..., R_k^(N)}        ▷ Threshold to be marked
3:     for all e ∈ P do
4:         if R_k^(e) ≥ ς_cut then
5:             e ∈ P_R                  ▷ Add to the set of marked elements
6:         end if
7:     end for
8:     P_R ← conform_refinement(P_R)    ▷ Conform marked elements
9: end procedure
```

6.1.3 Refinement

Refinement is the fourth step in any adaptivity algorithm, and some considerations due to the semi-structured characteristic of the triangular prism have to be taken into account. The refinement strategy adopted here is based on two-dimensional refinements for triangles and, after that, an extrusion is performed in the segment direction, being split if required. Thus, refinement is particularized to two directions: *horizontal* direction, where the triangles which conform top and bottom faces of the prisms are contained, and *vertical* direction, which is the extrusion direction of the prism. Refinements in this document are denoted as a number of two digits: the first digit is referred to the horizontal direction, while second digit is relative to the vertical direction. For the horizontal direction, different alternatives can be found: here, the well-known *red-green* algorithm is used, [125].

In short, when a triangle is marked to be refined, is set to *red*, which means that four new triangles are created joining the middle points of each edge. In the neighbor triangles, if they are not marked to be refined, they are set to *green*, which means that the middle point of the

Pseudocode 6.4 Marking strategy based on obtaining a set of elements to be refined which constitutes a given percentage of the total energy of the residuals

Require: conform_refinement($\mathcal{P}_\mathrm{R}$) ▷ Process of conformation to avoid hanging nodes
Require: order_residual($\mathcal{R}_k$) ▷ Order elements from highest to lowest $\mathcal{R}_k^{(m)}$
Input: $\mathcal{R}_k^{(m)}$ ▷ Evaluation of (6.6)
Input: θ ▷ Percentage to set amount of energy of $\mathcal{P}_\mathrm{R}$
Input: $\mathcal{P}$ ▷ Set of original elements (of size N)
Output: $\mathcal{P}_\mathrm{R}$ ▷ Set of elements marked to be refined (of size N_R)

1: **procedure** MARK_FIXEDENERGY($\mathcal{R}_k^{(m)}$,θ,$\mathcal{P}$,$\mathcal{P}_\mathrm{R}$)
2: $\varsigma_\mathrm{cut} \leftarrow \theta \cdot \sum_{m=1}^{N} \mathcal{R}_k^{(m)}$ ▷ Threshold to be marked in energy
3: $\mathcal{P}_\mathrm{ord} \leftarrow$ order_residual($\mathcal{R}_k$) ▷ Get an ordered set of elements
4: $N_\mathrm{ref} \leftarrow 1$ ▷ Initialize number of elements to mark.
5: **while** $\sum_{m=1, m\in\mathcal{P}_\mathrm{R}}^{N_f} \mathcal{R}_k^{(m)} \leq \varsigma_\mathrm{cut}$ **do** ▷ N_f is the size of $\mathcal{P}_\mathrm{R}$
6: **for all** $e \leq N_\mathrm{ref}$ **do**
7: $e \in \mathcal{P}_\mathrm{R}$ ▷ Add to the set of marked elements
8: **end for**
9: $\mathcal{P}_\mathrm{R} \leftarrow$ conform_refinement($\mathcal{P}_\mathrm{R}$) ▷ Conform marked elements
10: $N_\mathrm{ref} \leftarrow N_\mathrm{ref} + 1$ ▷ Increase the number of elements to refine.
11: **end while**
12: **end procedure**

Pseudocode 6.5 Slight variation of [6] where a variable threshold is defined to obtain a set of marked elements which conform a given percentage of the total energy of the residuals

Require: conform_refinement($\mathcal{P}_\mathrm{R}$) ▷ Process of conformation to avoid hanging nodes
Require: order_residual($\mathcal{R}_k$) ▷ Order elements from highest to lowest $\mathcal{R}_k^{(m)}$
Input: $\mathcal{R}_k^{(m)}$ ▷ Evaluation of (6.6)
Input: θ ▷ Percentage to set amount of energy of $\mathcal{P}_\mathrm{R}$
Input: $\mathcal{P}$ ▷ Set of original elements (of size N)
Output: $\mathcal{P}_\mathrm{R}$ ▷ Set of elements marked to be refined (of size N_R)

1: **procedure** MARK_SER($\mathcal{R}_k^{(m)}$,θ,$\mathcal{P}$,$\mathcal{P}_\mathrm{R}$)
2: $\varsigma_{\mathrm{cut,energy}} \leftarrow \theta \cdot \sum_{m=1}^{N} \mathcal{R}_k^{(m)}$ ▷ Threshold to be marked in energy
3: $\mathcal{P}_\mathrm{ord} \leftarrow$ order_residual($\mathcal{R}_k$) ▷ Get an ordered set of elements
4: $N_\mathrm{ref} \leftarrow 1$ ▷ Initialize number of elements to mark.
5: $\theta_\mathrm{R} \leftarrow 1$ ▷ Initialize threshold to mark elements
6: **while** $\sum_{m=1, m\in\mathcal{P}_\mathrm{R}}^{N_f} \mathcal{R}_k^{(m)} \leq \varsigma_{\mathrm{cut,energy}}$ **do** ▷ N_f is the size of $\mathcal{P}_\mathrm{R}$
7: $\theta_\mathrm{R} \leftarrow \theta_\mathrm{R} - 0.02$ ▷ Decrease threshold in each iteration
8: **for all** $e \leq N_\mathrm{ref}$ **do**
9: $e \in \mathcal{P}_\mathrm{R}$ ▷ Add to the set of marked elements
10: **end for**
11: $\mathcal{P}_\mathrm{R} \leftarrow$ conform_refinement($\mathcal{P}_\mathrm{R}$) ▷ Conform marked elements
12: **end while**
13: **end procedure**

edge neighbor with the red triangle is joined to the opposite vertex. Obviously, if some triangle has two neighbors which are red, this triangle is transformed into a *red* triangle, although some variations (the algorithm known as *red-green-blue*, [126]) may also be used. In terms of notation, when a prism is marked as red, it is noted with the number 2, while if it is marked as green, it is noted with the number 1. Finally, in the vertical direction there is only one possibility: to refine by the middle point in the edge, noted here with the number 1. If no refinement is applied in that direction, it is noted as 0.

With this notation, five different refinements appear, which are shown individually in Figure 6.2. Moreover, the semi-structured feature of the prism forces the application of a conformation that regularizes —i.e., without hanging nodes— the resulting mesh. This process consists of propagating the refinement horizontally and vertically, which means, e.g., that if the triangular faces of some prism are refined, the triangular faces of the neighbors on the top and the bottom have to be refined. When an element is marked to be fully refined, it will be noted as *21* (red refinement in the top and bottom triangular faces and refinement in the segment). Let's consider that only this element is marked to be refined. Then, horizontal neighbors (elements with top and bottom faces contained in the same plane as the element marked as *21*) will be marked as *11*, since triangles are marked as green and refinement in the extrusion direction forces all the elements in the same layer to be refined in the segment. For this reason, remaining neighbors of these *11* elements will be marked as *1*. Finally, bottom and top neighbors of the *21* elements will be marked as *20*, since no refinement in the segment is necessary but triangles have to be refined to keep regularity in the mesh. The last possible element are the neighbors of the *20* elements, which will be marked as *10*. An example of this situation is shown in Figure 6.3. All the geometries obtained after the refinement are geometrically admissible which means, in short, that two different elements in a triangle tessellation share and edge, a vertex, or nothing, [151]. Indeed, the problem of hanging nodes is not treated in this dissertation due to its complexity. Thus, the mesh obtained through this refinement will be conformal, or as it might be provided by a standard mesher.

Finally, two considerations have to be made for the refined meshes. The use of green triangles (*1x*) may lead to *holes* or *eyes* in the mesh if, given one vertex shared by many elements (e.g., eight elements), some

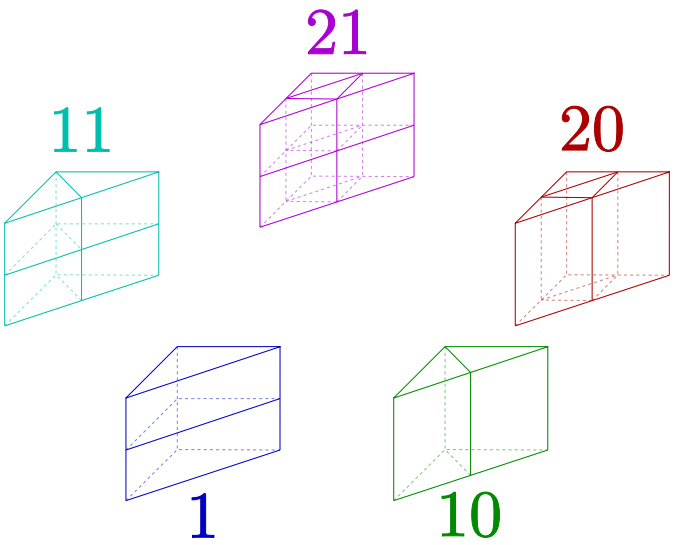


Figure 6.2: Each possible refinement for triangular prisms. Only one case out of three for *10* and *11* cases are shown for brevity

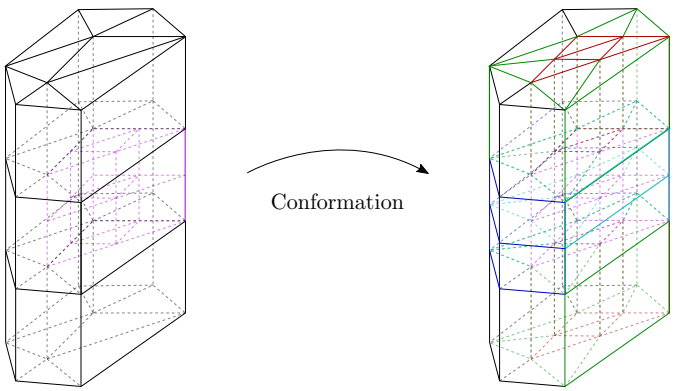
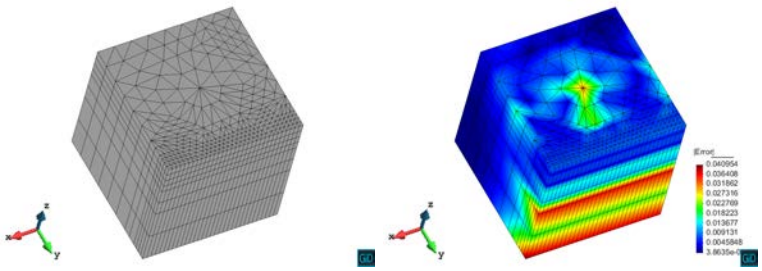


Figure 6.3: An example of application of conformation process in refined meshes

elements are marked as *1x* being this one vertex the vertex whose angle is divided. Then, if the error is accumulated in this point, there is no backward step (if the elements inside the hole are further refined, the number of elements which share the same vertex is the same). One solution to prevent this situation is to introduce an accumulator to all the vertices in the mesh and count the number of elements which have this vertex. With this accumulator, if the situation explained before occurs, the element is marked as *2x*. Moreover, if no criteria for good quality meshes are imposed, some elements with green refinement in the triangular faces might lead to very deformed triangles in the top and bottom faces which can increase the error of a given mesh (although these elements would be refined in the next adaptivity step) and conditioning of the resulting mesh is worsened. To avoid that, when an element is marked as *11* or *10*, the two angles associated to the opposite vertex of the split edge are checked and if they are lower than a given quantity (23 degrees has been proven in experiments as a reasonable threshold), they are marked as *21*. An example of these two criteria to generate a mesh with good quality is shown in the Section 6.2.

Regarding DDM, when this refinement is applied individually to different subdomains, it is well-known, [85], that a rule of thumb has to be kept between elements, i.e, when the size of elements on a given interface is very different, a non-negligible error is located on the interface. Although in Section 5.4.1 it is observed that this aspect ratio seems not to be critical in the error provided by the non-conformal interface, it has to be noted that adaptive meshes with triangular prisms may lead to very unbalanced non-matching meshes on the interface of, e.g., sixteen elements from one subdomain communicating with only one element from the other.

To avoid that, a threshold on the aspect ratio of the elements belonging to the interface between subdomains may be imposed if the so-called ROT criterion is applied. Essentially, this criterion compares the size of the neighbor elements in both sides of the interface. If the difference is bigger than a given threshold, the same refinement to all the neighbor elements are applied. Note that, if this criterion is not applied, the error due to the non-conformality will be similar in both subdomains. This may lead the adaptivity algorithm to refine all the elements belonging to the interface, which yields a finer mesh with the same non-conformality between the subdomains. This effect has been simulated in Table 5.9



(a) Mesh **(b)** Error

Figure 6.4: Presence of anomaly (*eye*) in the adaptive mesh

where it can be observed that the error is not decreased by this refinement.

Note that if the ROT criterion is applied for original matching interfaces, the effect is that conformality is retained on the interface for all the iterations of the refinement algorithm. An example of application of this criterion is shown in Sec 6.2.3.

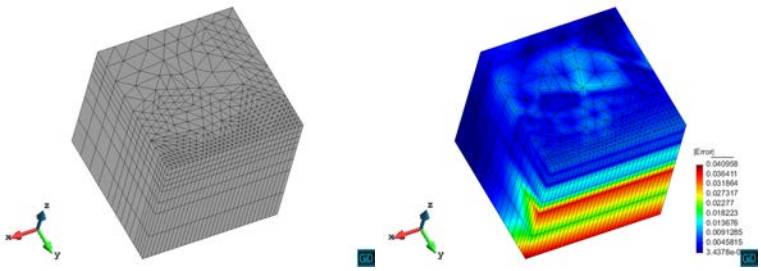
6.2 Verification through MMS

Following the procedure shown in previous chapters, MMS is going to be used to test the adaptivity algorithm presented in Section 6.1. A cube of length 1 m and working frequency of 50 MHz is used as the problem to be solved. In this section, the following goals are pursued: *i)* to validate the adaptivity algorithm; *ii)* to test the five different marking strategies introduced in this dissertation; *iii)* to study the effect of DDM with conformal and non-conformal meshes on the interface between subdomains.

6.2.1 Considerations with green triangles

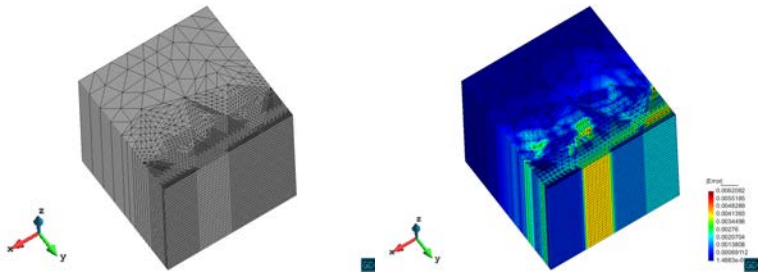
The effect of the two criteria introduced in the process of conformation for *1x* elements is exposed. First, an example of use for the first criterion, related to the possible appearance of *holes* in the mesh if no criteria is applied, is shown in Figure 6.4 and 6.5. Here, it can be seen that some error is accumulated near the center of the top and bottom faces because too many elements converge to the same point. In Figure 6.5 the treatment explained in the previous section is applied, disappearing such error increase in an abnormal area.

Now, an example of use for the second criterion (related to the min-



(a) Mesh **(b)** Error

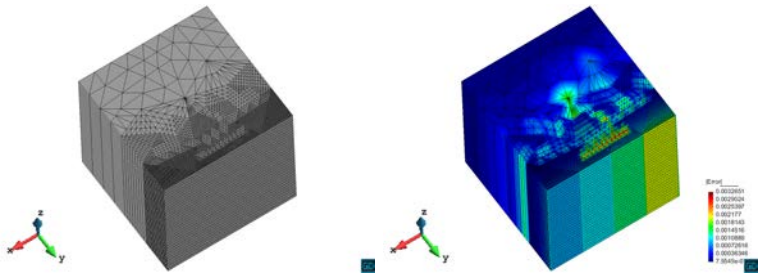
Figure 6.5: Treatment of *eyes* in the adaptive mesh



(a) Mesh **(b)** Error

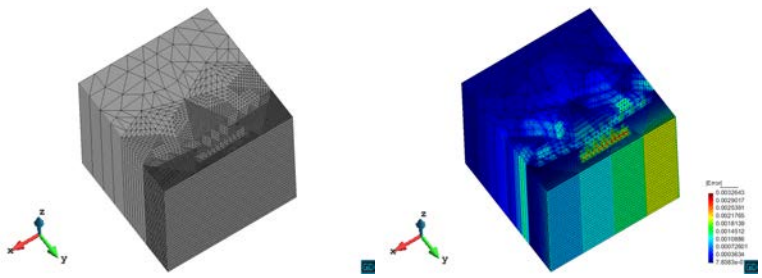
Figure 6.6: Presence of distorted elements in the adaptive mesh

imum angle allowed for an element) is shown in Figure 6.6. where no limitations to the quality of the mesh generated are applied. Here, all the elements marked as *1x* are forced to have a maximum angle of 0.4 rad in the vertex from which the division is made. With this, the number of elements is increased but no strange error areas are present, as shown in Figure 6.7. However, this adjustment for not having specially distorted elements produces an error due to two *holes* that has not been treated. If the criterion explained before is applied here, in Figure 6.8 the error distribution and the mesh are shown. The error shown in that figure is due to the discretization applied in the areas of higher error, refined in next iterations. If no criteria (*holes* and distortion of elements) is applied, strange meshes with strange distribution of errors are produced, as shown in Figure 6.6 and 6.4.



(a) Mesh **(b)** Error

Figure 6.7: Adaptivity with a criterion for not allowing elements with an angle lower than 0.4 rad



(a) Mesh **(b)** Error

Figure 6.8: Adaptivity with criteria for not allowing neither holes nor elements with an angle lower than 0.4 rad

Table 6.3: θ parameter for verification with MMS

$\mathbf{E}_{\text{MMS}}$	y^9
Quantile	0.3
Maximum	0.05
Fixed-Energy	0.9999
SER	0.9

6.2.2 Effect of marking strategies

To analyze the performance of a semi-structured element as the triangular prism, the polynomial

$$\mathbf{E}_{1,\text{MMS}} = \begin{Bmatrix} y^9 \\ y^9 \\ y^9 \end{Bmatrix}, \tag{6.8}$$

is going to be used since it is clearly outside of the space of basis functions and it produces stronger fields in a specific area of the subdomain to be solved. The same electromagnetic domain defined in the previous section is used.

In the following, the five marking strategies are tested, and the ROT criterion is applied for illustration purposes. First, the error obtained for each marking strategy is shown in Figure 6.9. The error computed here is the maximum error in the mesh, introduced with

$$\varsigma_{\max} = \frac{\max\{\|\mathbf{E}_{\text{FEM}} - \mathbf{E}_{\text{MMS}}\|_2\}}{\|c_2\left(\mathbf{E}_{\text{MMS}}, \mathbf{E}^*_{\text{MMS}}\right)\|_2}, \tag{6.9}$$

since the estimator is computed through absolute values of the field. Thus, the maximum error defined in (6.9) (where $\mathbf{E}_{\text{MMS}}$ in the denominator is the same which maximizes the numerator) is used in Figure 6.9. Marking strategies have been used with the parameters shown in Table 6.3. For the uniform refinement, unstructured meshes have been used for the top and bottom faces.

The adaptive refinement improves the results obtained with uniform refinement but not as much as it should be. It is not the focus of this dissertation to build adaptive meshes to achieve exponential rates of convergence but also as illustration of independent adaptivity in different

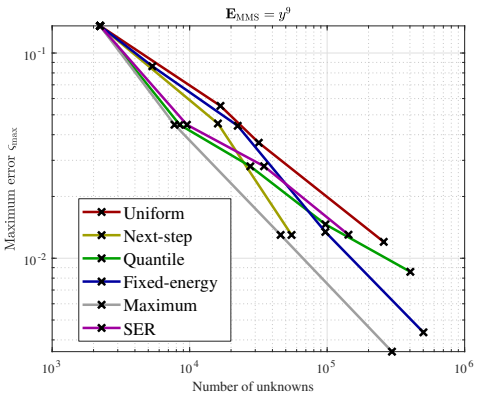
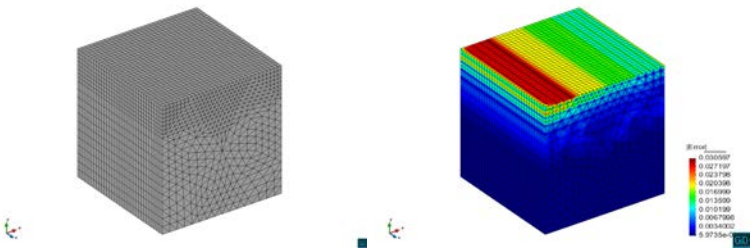


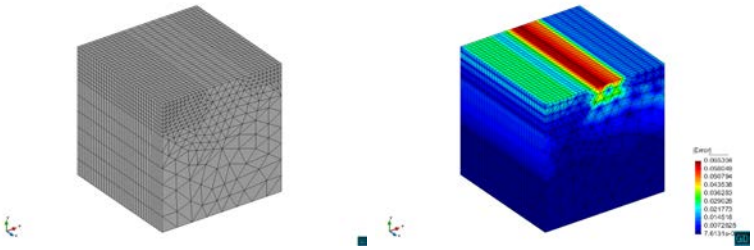
Figure 6.9: Results with different marking strategies using (6.9)

subdomains. However, this lack of improvement might be expected due to the semi-structured nature of the prism and generality of the adaptivity algorithm introduced here, not suitable for this problem. To better illustrate this, meshes and error for different marking strategies are included in Figs. 6.10, 6.11, 6.12, 6.13, and 6.14, where three rounds of refinement have been introduced. For this particular case, the longitudinal direction of the prism should be oriented along the Y axis while the elements should be marked only with the so-called *1* refinement in Section 6.1.3. Moreover, some differences between marking strategies may be found, although the performance achieved depends strongly on the θ parameter and no definitive differences in performance are obtained (note that the magnitude of the error field shown in the figures is on the same order). Strategies based on energy of the residual (fixed-energy and SER) show a more clustered refinement in the areas of stronger error, while the maximum strategy shows a more uniform refinement. The strategy based on the number of elements with higher residual can lead to some irregular meshes (since an element with much lower residual than another takes the same refinement as the element of highest residual). Finally, next-step marking strategy seems to provide a more structured mesh because of the first step (in which all the mesh is refined) necessary to estimate subsequent adaptivity steps.



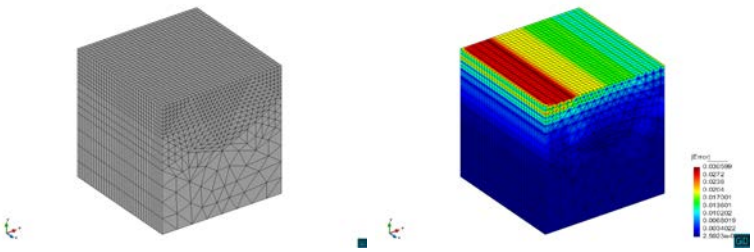
(a) Mesh **(b)** Error

Figure 6.10: Approximation of the monomial using the next-step marking strategy



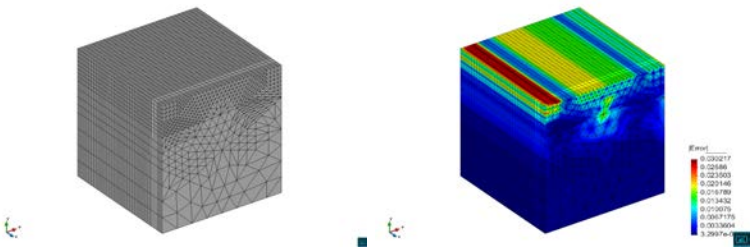
(a) Mesh **(b)** Error

Figure 6.11: Approximation of the monomial using the quantile marking strategy



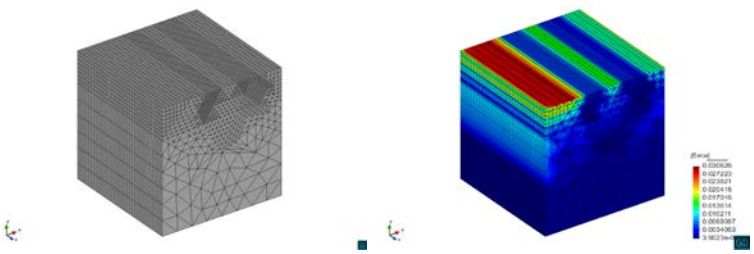
(a) Mesh **(b)** Error

Figure 6.12: Approximation of the monomial using the maximum marking strategy



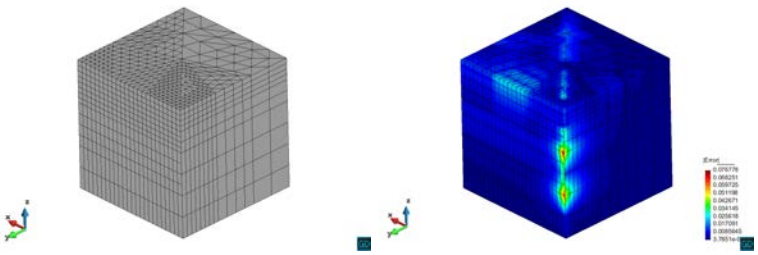
(a) Mesh **(b)** Error

Figure 6.13: Approximation of the monomial using the fixed-energy marking strategy



(a) Mesh **(b)** Error

Figure 6.14: Approximation of the monomial using the SER marking strategy



(a) Mesh **(b)** Error

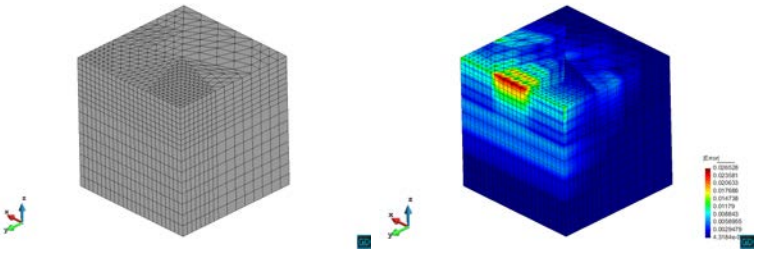
Figure 6.15: Approximation of y^5z^5 with the fixed-energy marking strategy and DDM

6.2.3 Adaptivity with DDM

In this section, two goals are pursued: *i)* to check the accuracy of DDM with meshes generated with the adaptivity algorithm, and *ii)* to test the effect of the interface between subdomains on the distribution of the error. The domain to be solved is the same as in Section 5.1, while the polynomial used is $\mathbf{E}_{\text{MMS}} = (y^5z^5, y^5z^5, y^5z^5)$. The division into two subdomains is introduced in a user-driven and conformal way. The mesh generated and distribution error with the fixed-energy strategy is shown in Figure 6.15.

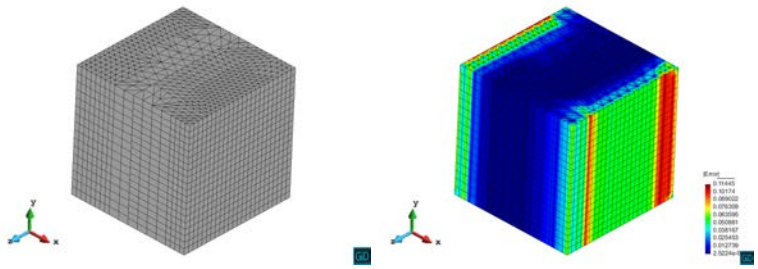
If another iteration of the adaptivity algorithm is performed, the elements to be refined belong to the interface between subdomains. Since this error is present in both subdomains, this iteration will keep the aspect ratio and the non-conformality between subdomains and the error will not be reduced. Three solutions to this problem may be applied: *i)* to detect the difference in the aspect ratio and refine the bigger elements on the interface; *ii)* to penalize the residual obtained for the elements which belong to an interface between subdomains; and *iii)* to apply the ROT criterion which, in this case, retains the conformality in the whole problem. For illustration purposes, the ROT criterion is introduced here generating the results included in Figure 6.16, where it can be seen that no special distribution of error is shown due to the presence of a discontinuity between subdomains.

Finally, an example of application to an original non-conformal DDM is introduced in the following. Now, a cube split in two subdomains as in Section 5.3 is simulated, where the mesh is non-conformal but com-



(a) Mesh **(b)** Error

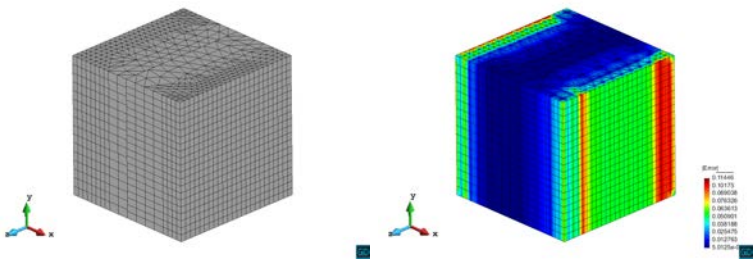
Figure 6.16: Approximation of y^5z^5 with the maximum marking strategy and DDM with the ROT criterion



(a) Mesh **(b)** Error

Figure 6.17: Approximation of x^7 in a non-conformal mesh DDM using the maximum marking strategy

parable in size between the two subdomains, and polynomial introduced in MMS is $\mathbf{E}_{\text{MMS}} = (x^7, x^7, x^7)$. Since the distribution of this function is symmetrical for both subdomains, very similar refinements should be obtained. However, if any of the three solutions commented above is introduced, the refined mesh follows a strange pattern trying to reduce the error on the interface, as shown in Figure 6.17. For illustration purposes, if the ROT criterion is applied with a threshold for the aspect ratio of 20%, a more consistent mesh is obtained, as it is shown in Figure 6.18.



(a) Mesh **(b)** Error

Figure 6.18: Approximation of x^7 in a non-conformal mesh DDM using the maximum marking strategy and the *rule of thumb* criterion

6.3 Real problems

Once the adaptivity algorithm has been validated in terms of accuracy when DDM is involved, some real problems are tested in the following to harness the potential of this technique. The main objective of this section is to study the effect of the non-conformal interfaces generated when each domain is refined independently.

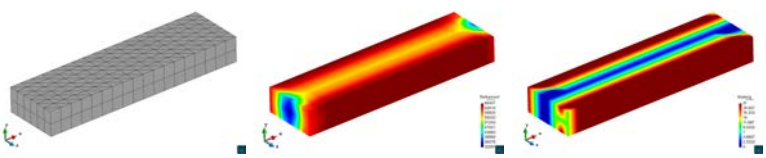
6.3.1 WR-90 waveguide

In this section, the effect of non-conformality between subdomains in the estimator introduced in Section 6.1.1 is studied. A WR-90 waveguide with length $l = 1\lambda$ is considered as the problem to be refined, with a working frequency of $f = 7.5\,\text{GHz}$. First, a structured mesh is introduced as discretization to show clearly the effect of refinement. Mesh, estimator from (6.6), and marking of the elements to be refined in the first iteration are shown in Figure 6.19. Here, marking follows the convention shown in Figure 6.2, so elements to be refined are closer to the electric walls. The second iteration is introduced in Figure 6.20, where it can be seen that the error is decreased in the elements refined previously, so now the main contribution to this error is in the middle of the waveguide, where the elements are marked to be refined. If a new iteration is applied, the same effect can be observed.

Now, the same study is introduced for an unstructured mesh. Here, the effect is the same as with the structured mesh but the shape of the estimator is not as clear as shown in Figs. 6.22, 6.23 and 6.24. In order to detect possible anomalies in the estimator, an error ς_{wg} defined as

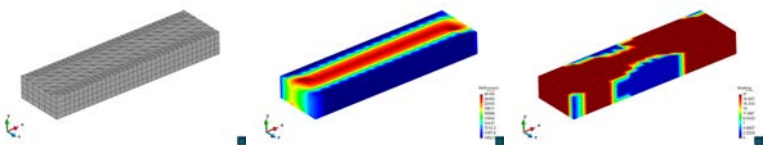
$$\varsigma_{\text{wg}} = \boldsymbol{\nabla} \times \mu_{ri}^{-1}(\boldsymbol{\nabla} \times \mathbf{E}_h) - k_0^2 \varepsilon_{ri} \mathbf{E}_h, \tag{6.10}$$

where $\mathbf{E}_h$ is the error field, $\mathbf{E}_h = \mathbf{E} - \mathbf{E}_{\text{anal}}$, shown in Figure 6.25. It can



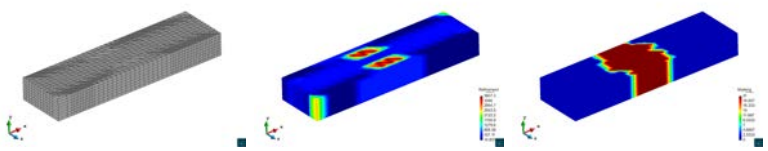
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.19: First iteration for a structured mesh of a WR-90 waveguide



(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.20: Second iteration for a structured mesh of a WR-90 waveguide



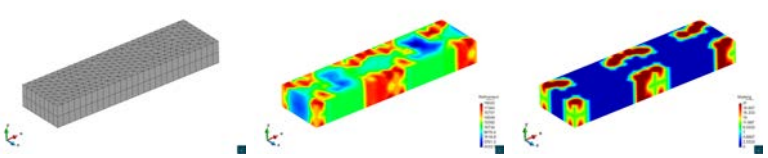
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.21: Third iteration for a structured mesh of a WR-90 waveguide

be seen that the agreement in the shape of the error and the estimator is quite good.

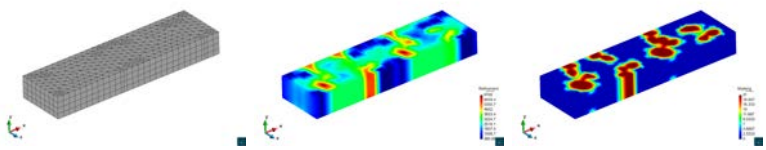
In the following, a conformal division into two subdomains is applied to the same mesh as above generating two 0.5λ sections. Results obtained from the solution of this DDM problem are shown in Figs. 6.26, 6.27 and 6.28. The same refined meshes than in the case of not using DDM are generated, as it was expected.

Finally, the mesh is slightly changed to obtain a non-conformal mesh



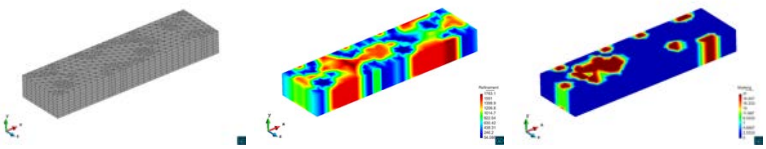
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.22: First iteration for a unstructured mesh of a WR-90 waveguide



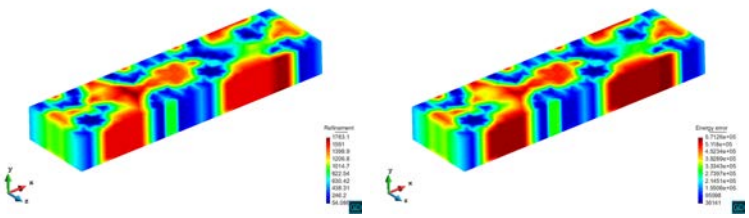
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.23: Second iteration for a unstructured mesh of a WR-90 waveguide



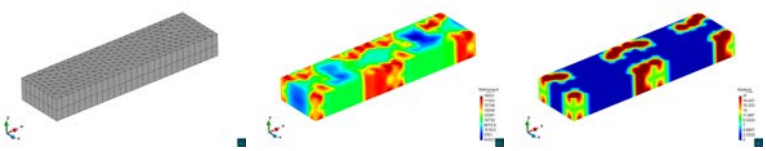
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.24: Third iteration for a unstructured mesh of a WR-90 waveguide



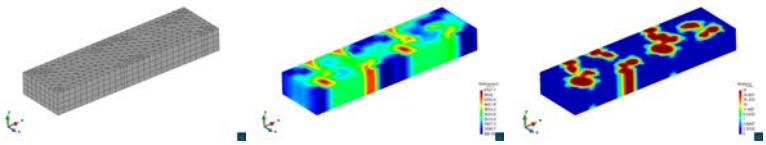
(a) Estimator **(b)** Error

Figure 6.25: A comparison between the estimator and the error in the third iteration of refinement



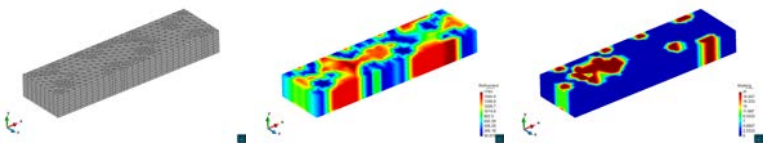
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.26: First iteration for a unstructured mesh of a WR-90 waveguide with DDM



(a) Mesh **(b)** Estimator **(c)** Marking

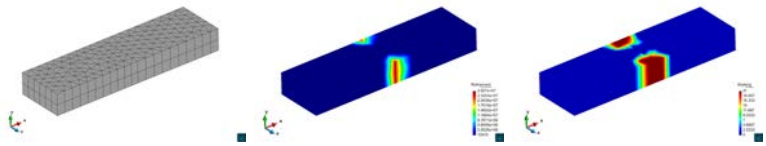
Figure 6.27: Second iteration for a unstructured mesh of a WR-90 waveguide with DDM



(a) Mesh **(b)** Estimator **(c)** Marking

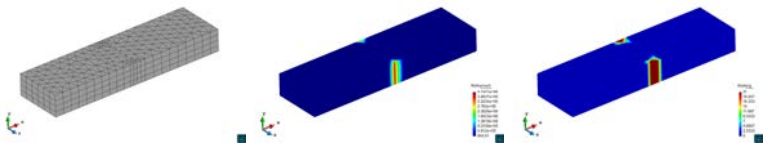
Figure 6.28: Third iteration for a unstructured mesh of a WR-90 waveguide with DDM

solving the same electromagnetic problem. Three iterations of refinement are shown in Figs. 6.29, 6.30 and 6.31, obtaining interesting results. Note that the scale of the estimator is much higher than the scale shown in the previous experiments. First, it is observed that the error is accumulated on the interface between subdomains so the refinement is focused on the interface. This refinement leads to a reduction of the value of the estimator with the number of adaptivity iterations. Indeed, if a uniform refinement is applied to the same original problem shown in Fig. 6.29, similar value of the estimator is obtained after three rounds as shown in Figure 6.31 as expected. Indeed, if the value of $|s_{21}|$ is taken into account for each iteration, it evolves from 0.999560 in the first round, to 0.999956 in the second iteration and 0.999992 in the third round. Indeed, this last value is comparable to the error obtained for conformal DDM. However, this evolution is not obtained with uniform refinement: 0.999560 in the first iteration, 0.999861 in the second round and 0.999981 in the third iteration with a much higher number of unknowns. As it is shown in the Chapter 5, the uniform refinement in a non-matching interface does not provide a decrease in the error due to the non-conformality, so adaptive meshes constitute a promising approach to reduce this error.



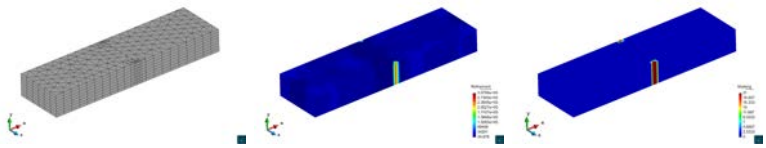
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.29: First iteration for a unstructured mesh of a WR-90 waveguide with non-conformal DDM



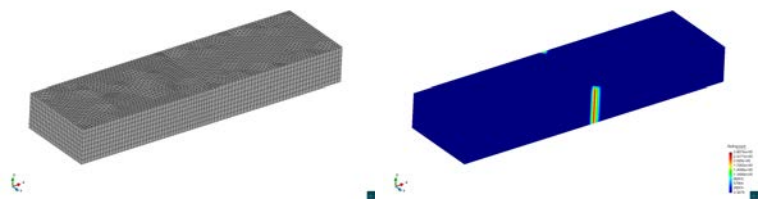
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.30: Second iteration for a unstructured mesh of a WR-90 waveguide with non-conformal DDM



(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.31: Third iteration for a unstructured mesh of a WR-90 waveguide with non-conformal DDM



(a) Mesh **(b)** Estimator

Figure 6.32: Mesh and estimator of the error for third iteration of uniform refinement with non-conformal DDM

Therefore, this test bench is very interesting for the application of h adaptivity with non-conformal DDM: first, it is seen that non-conformal interfaces introduce an additional error which will make the adaptivity algorithm refine in the non-conformal interface. Although the error is reduced with adaptive refinement, this is not the point of introducing DDM for adaptivity. It is more convenient to take into consideration the error introduced by the non-conformality interface and, if this error becomes too high, to introduce some adaptive refinements on the interface. Thus, a good solution is to penalize the error obtained by the estimator in the vicinity of the non-matching interfaces. Another possible solution is to use the ROT criterion to keep the aspect ratio below a given threshold. Although the error due to the non-conformality interface is not very sensitive to the aspect ratio, successive iterations of the adaptivity algorith may lead to aspect ratios of 16, 64... which means that only one element from one subdomain is the neighbor for 64 elements from the other subdomain.

6.3.2 L-shaped waveguide

One of the easiest ways to introduce a singularity in an electromagnetic problem is to bend a waveguide along its transverse face (E-plane) where the electric field is maximum. The waveguide to be bent is a WR-90 waveguide (i.e., dimensions of the waveport are $0.02286\,\mathrm{m}\times0.01016\,\mathrm{m}$) as shown in Figure 6.33. This problem is divided conformally in five subdomains Ω_1, Ω_2, Ω_3, Ω_4, and Ω_5 through input data, and working frequency is $f = 7.5\,\mathrm{GHz}$, within the operational bandwidth of the waveguide.

Four different experiments are performed. For illustration purposes, the ROT criterion is used to maintain conformal interfaces in the first three experiments: *i)* only h adaptivity; *ii)* h adaptivity with uniform p distribution; and *iii)* h adaptivity with p refinement with the estimator from Section 6.1.1. Last experiment performed here is h adaptivity with a strong penalization for the residual of the elements which belong to the non-conformal interfaces, and the ROT criterion is not applied. In practice, elements which belong to the non-conformal interfaces are not marked to be refined (although they are refined to obtain an admissible mesh). The maximum marking strategy with $\theta = 0.65$ is used for all the experiments.

Results of mesh, estimator and marking for the first experiment are included in Figs. 6.34, 6.35 and 6.36, while the evolution of the field for each iteration of the algorithm is included in Figure 6.37. The singularity is better represented with each iteration of the adaptivity algorithm, while no marking is introduced on the interfaces between subdomains. For the sake of comparison, evolution of the electric field with uniform refinement is shown in Figure 6.38.

Then, uniform p distribution with $p = 3$ in subdomains Ω_1 and Ω_5 and partially uniform p distribution in subdomains Ω_2 and Ω_4 is introduced as shown in Figure 6.39, while same marking strategy is used. It is shown in Figs. 6.40, 6.41 and 6.42 that no special issues appear when using different p within the same subdomain, refining elements in the singularity which is the most sensitive part to refinement and obtaining a similar approximation of the electric field as shown in Figure 6.43. Related to the estimator, it is also seen that the lowest value decreases due to the introduction of a higher p.

Next experiment included is introducing non-uniform p refinement within subdomains Ω_1, Ω_2, Ω_4 and Ω_5. Maximum order allowed is $p =$

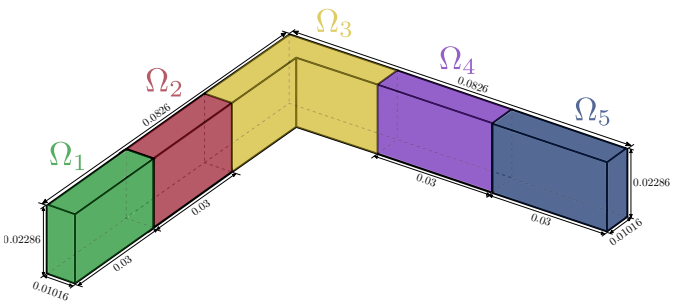
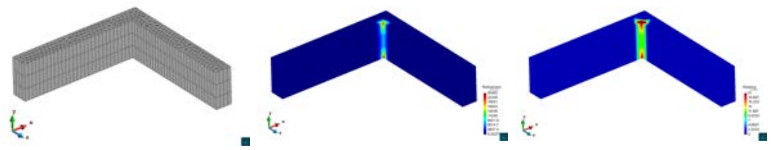
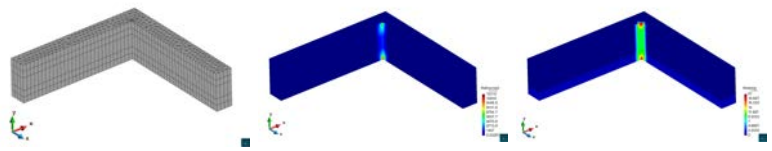


Figure 6.33: WR-90 bend waveguide used in Section 6.3.2 as problem to be solved



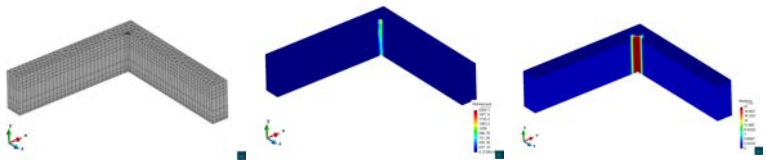
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.34: First iteration for L-shaped problem with h adaptivity and conformal DDM



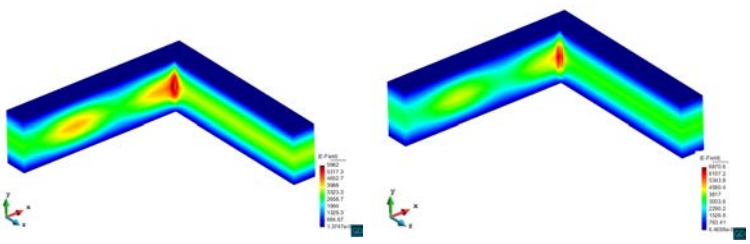
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.35: Second iteration for L-shaped problem with h adaptivity and conformal DDM

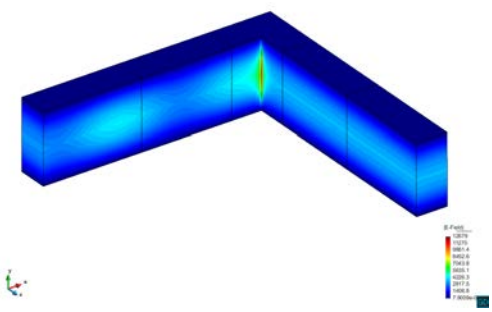


(a) Mesh (b) Estimator (c) Marking

Figure 6.36: Third iteration for L-shaped problem with h adaptivity and conformal DDM

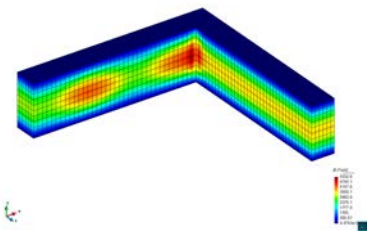


(a) First iteration (b) Second iteration

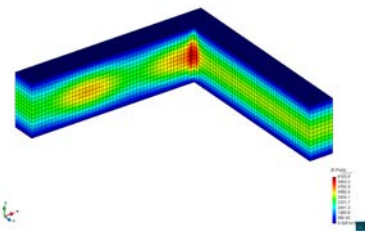


(c) Third iteration

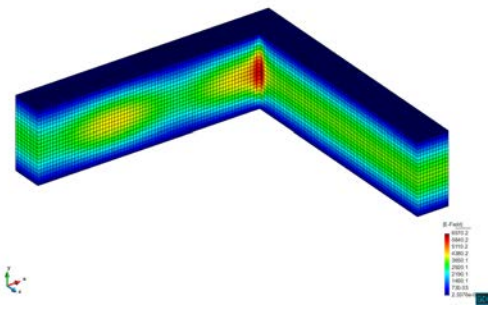
Figure 6.37: Evolution of the electric field for h refinement in a L-shaped domain



(a) First mesh



(b) Second mesh



(c) Third mesh

Figure 6.38: Evolution of the electric field for uniform refinement in a L-shaped domain

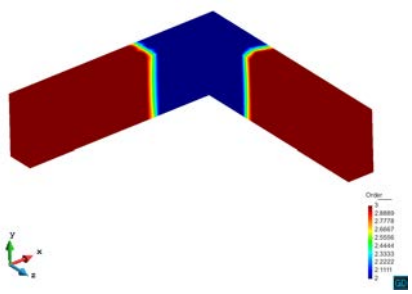
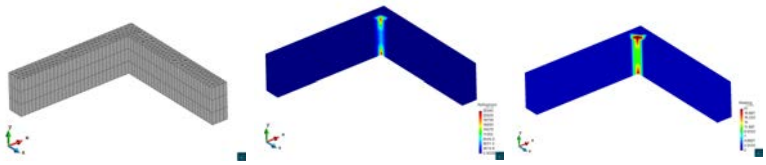
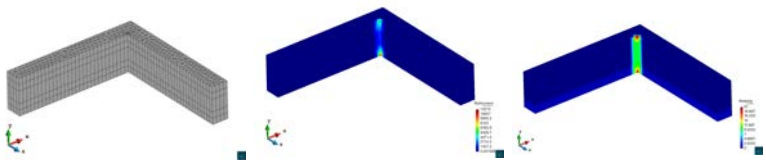


Figure 6.39: Order for uniform p refinement test



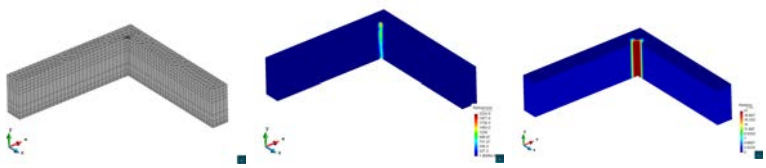
(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.40: First iteration for L-shaped problem with h adaptivity, p distribution and conformal DDM



(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.41: Second iteration for L-shaped problem with h adaptivity, p distribution and conformal DDM



(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.42: Third iteration for L-shaped problem with h adaptivity, p distribution and conformal DDM

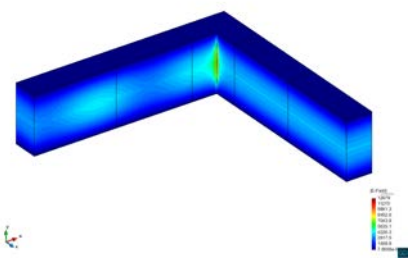
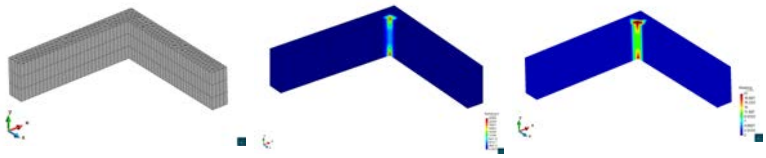
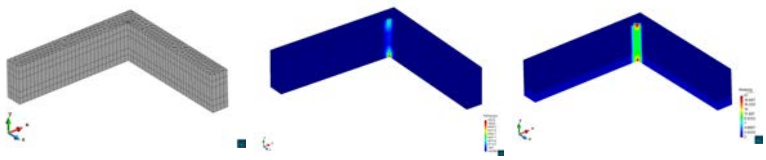


Figure 6.43: Electric field for h refinement and different p in a L-shaped domain after adaptivity refinement



(a) Mesh (b) Estimator (c) Marking

Figure 6.44: First iteration for L-shaped problem with h adaptivity, p refinement and conformal DDM



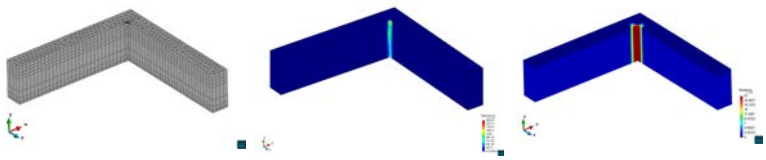
(a) Mesh (b) Estimator (c) Marking

Figure 6.45: Second iteration for L-shaped problem with h adaptivity, p refinement and conformal DDM

3 in Ω_1 and Ω_5 whereas $p = 4$ is set in Ω_2 and Ω_5. The estimator used to mark elements to be p refined is the same estimator introduced in Section 6.1.1. No special issues regarding h adaptivity are present when using p refinement, as shown in Figs. 6.44, 6.45 and 6.46, and p refinement is introduced for those areas with bigger error within each subdomain, as shown in Figure 6.47. This experiment is included to show that with a non-uniform p refinement the electric field obtained is correct, as depicted in Figure 6.48, but no special accuracy is expected to be obtained since non-uniform p refinement is taken into account in the adaptivity algorithm.

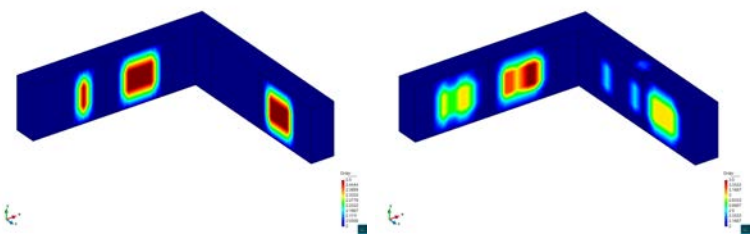
Finally, an experiment disabling the ROT criterion is introduced next. Only h refinement is applied. Evolution of the electric field is included in Figure 6.49. Similar results for the representation of the electric field in the singularity are obtained but a discontinuity appears between the subdomains which does not strongly affect the whole solution.

A semi-analytic solution for the scattering matrix can be obtained

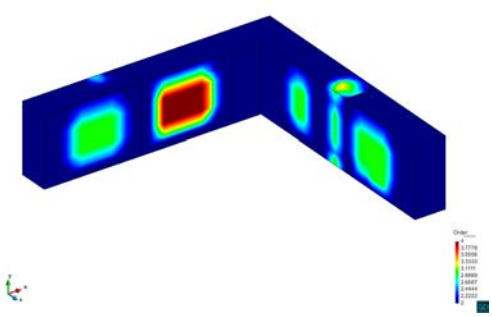


(a) Mesh **(b)** Estimator **(c)** Marking

Figure 6.46: Third iteration for L-shaped problem with h adaptivity, p refinement and conformal DDM



(a) First iteration **(b)** Second iteration



(c) Third iteration

Figure 6.47: Evolution of the p refinement for hp refinement in a L-shaped domain

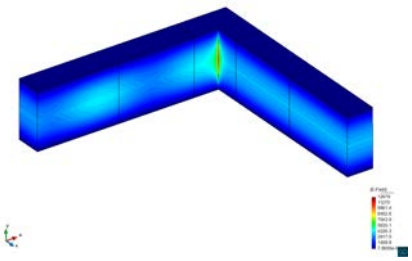
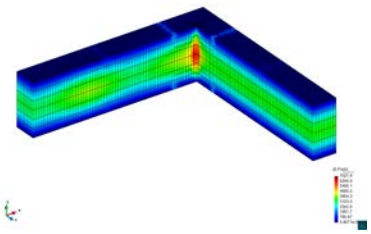
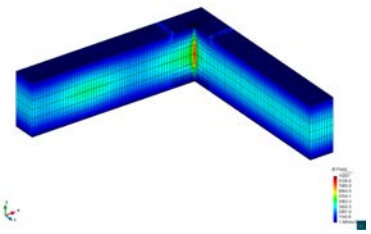


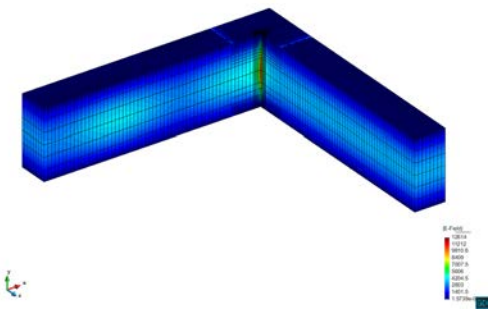
Figure 6.48: Electric field for h adaptivity and p refinement in a L-shaped domain after adaptivity refinement



(a) First mesh



(b) Second mesh



(c) Third mesh

Figure 6.49: Evolution of the electric field for h adaptivity without the *rule of thumb* criterion in a L-shaped domain

through modal analysis as introduced in [152] [1]. A comparison of the error in the approximation of scattering parameters, computed as

$$\varsigma_{\mathrm{s}} = \frac{1}{4}\sum_{i=1}^{2}\sum_{j=1}^{2}|s_{\mathrm{ij,anal}} - s_{\mathrm{ij,FEM}}|, \tag{6.11}$$

is included in Figure 6.50. Results with uniform refinement are included for comparison purposes. It is observed that results obtained with conformal interfaces between subdomains are as good as expected since the accuracy is not affected by the introduction of conformal DDM. Also, worse performance is obtained for the results with higher p as it would be expected since no special treatment is applied. Indeed, higher p would help in this particular problem exclusively for smooth field areas far away from the singularity. Regarding the experiment with the ROT criterion disabled, it is observed that ς_{s} remains flat with the number of iterations. Note that, even although the singularity is being approximated with more fidelity, the aspect ratio is worse on the interface between subdomains. Moreover, the non-conformality is close to the singularity which may lead to additional errors.

In Figure 6.51, the maximum value obtained in the singularity with the number of unknowns is represented as a measurement related to the adaptivity. In this set of results, the best performance is obtained when the ROT criterion is disabled. This might be expected since the value of ς_{s} low enough for not compromising the solution in the whole problem.

6.4 Conclusions

An adaptivity algorithm for triangular prisms has been introduced in Section 6.1, where each step is detailed: first, an estimator inspired by [153] is included; then, five different marking strategies are presented and, finally, conformation techniques to construct admissible meshes are included when marked elements are refined. The algorithm is tested with MMS, introducing some criteria related to the quality of the mesh. Moreover, some considerations related to DDM are introduced in Section 6.2.3, and the so-called here ROT criterion is introduced to keep the aspect ratio below a given threshold on a non-conformal interface. Although the aspect ratio is not a key factor as proven in Section 5.4.1,

[1] The author wants to thank Dr. Sergio Llorente-Romano for his help in this analysis.

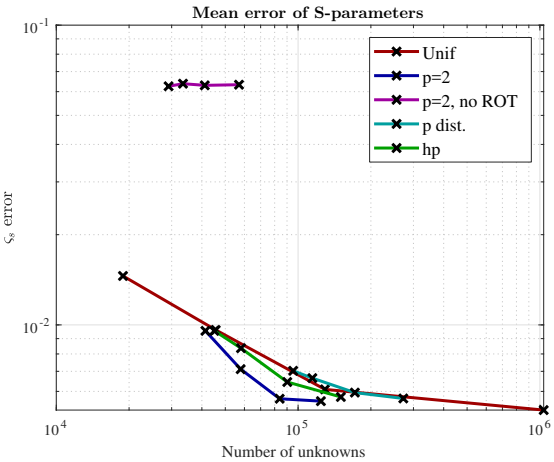


Figure 6.50: Error ς_s with all the tests

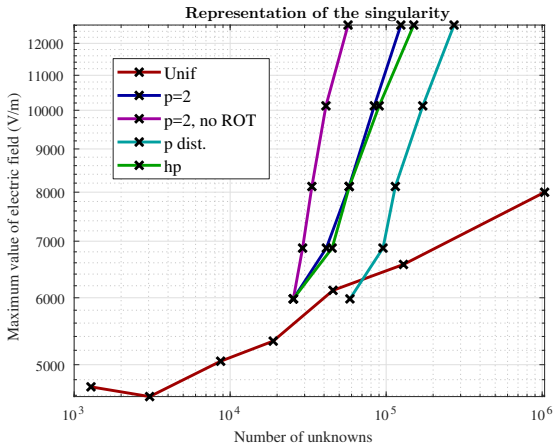


Figure 6.51: Maximum value of the field at the singularity with all the tests

the adaptive meshes with triangular prisms may lead to very unbalanced non-matching meshes on the interfaces.

Real problems are tested in Section 6.3. First, a WR-90 waveguide is taken as reference to assess the quality of the estimator in wave propagation problems. As it might be expected, the same refined mesh is generated if the adaptivity algorithm is applied to the same mesh independently of using DDM to solve the problem. Regarding non-conformal DDM, the adaptivity algorithm is forced to mark the elements belonging to the interface due to the error present on the interface. Thus, special consideration has to be taken in these cases. A good solution seems to penalize the error obtained by the estimator in the elements close to the interface. Moreover, although it is not the point of introducing adaptivity for DDM, adaptive refinements on the non-conformal interface seem to reduce the error unlike uniform refinements.

Secondly, an L-shaped domain is used to introduce a singularity in a WR-90 waveguide, and a division into five subdomains is applied. Four different experiments are performed, three keeping conformality on the interfaces between subdomains for illustration purposes: h adaptivity; h adaptivity with a given p distribution; h adaptivity with p refinement; and finally, one experiment with h adaptivity without applying the ROT criterion. The main conclusions drawn from the first three experiments are that the error in the scattering parameters is reduced with h adaptivity with the number of unknowns if it is compared with an uniform refinement, and that the increase in p to some elements does not reduce the error in this particular case. This is expected since no special techniques for p refinement have been introduced yet and, in this case, higher p elements would help only in smooth field areas close to the waveports. With respect to the last experiment, the error of S-parameters is quite flat with the number of iterations. This may be due to the high aspect ratio present on the non-conformal interface, which grows geometrically with the number of iterations. However, if the representation of the singularity is measured with the maximum value of the field in the problem, it has to be noted that the last experiment offers the best performance. Further studies will be done in the future regarding this problem and some solutions are proposed in Section 7.2.

CHAPTER 7

CONCLUSIONS AND FUTURE LINES

7.1 Conclusions

The conclusions that are obtained in this dissertation can be structured into three blocks: *i)* related to the basis functions; *ii)* relative to the introduction of non-conformal and non-overlapping DDM for a number of finite element shapes; *iii)* related to the introduction of *hp* meshes refined with *h* adaptivity into DDM.

Apart from these big three blocks, the formulation used along this dissertation is presented in Chapter 2 with special focus on the non-overlapping and non-conformal DDM implemented; and a parallel implementation of DDM is proposed in Chapter 3. Regarding this implementation, a three-level parallelization is introduced: *i)* at the algorithm level, with DDM, at the process level, using MPI, and at the thread level with OpenMP. Thorough explanations about the introduction of DDM in an OOP paradigm are provided, being helpful to fully understand the details of this technique.

In Chapter 4, two families of basis functions are introduced. The first family follows a sound, mathematically speaking, procedure to obtain basis functions as the dual basis of the DOFs introduced in [21,22]. These DOFs have to be discretized, leading to two different assembly strategies which are compared in Section 4.4: the so-called *vc* strategy is better when simplices are taken into consideration, while the *vq* strategy has a better performance for structured elements. This family is particularized for triangular prisms and a comparison with other families present in the literature is included, showing good results in the condition number of

FEM matrices. With respect to the second family, [26], it is included for hexahedra, tetrahedra and triangular prisms to ease the p refinement used in the third block of conclusions. Then, MMS is used to verify both families introducing a procedure that is used along this dissertation: first, monomials which are inside the space of functions are tested, obtaining results close to machine precision; and then, smooth exponential complex functions are introduced to generate convergence results which follow straight lines predicted by the theory. This procedure is quite convenient to introduce new techniques in a FEM code since the monomals offer a binary test of verification: if some error is introduced at some point in the code, the verification with monomials gives a non-null result, [13]. Finally, since dispersion error is a limiting factor for large scale problems, some preliminary results related to the phase error are included since no special difference is appreciated between the three different shapes.

With respect to the introduction of non-conformal and non-overlapping DDM, a thorough study regarding verification and validation is detailed in Chapter 5. First, a variation of typical MMS is introduced to check the formulation; then, the eigenspectra is obtained for a number of different combinations to test the convergence of the iterative solver; following this, MMS as used in the Chapter 4 is employed to verify the whole DDM implementation; and, finally, some real problems covering wave propagation, different dielectric media, radiation, high quality factor filters and RCS computations are addressed.

In the first step, direct solvers are used to solve the surface problem in order not to mask the numerical results with the accuracy of the iterative solver. Similar performance as in Chapter 4 is obtained. Regarding the eigenspectra study, the introduction of different elements has not been studied in the literature and, as expected, no additional error is introduced when conformal interfaces are studied. From these results, SOTC are expected to show a better performance in the convergence of the iterative solver than FOTC since the eigenvalues are more clustered around the center of the unit circle. In both cases, all the eigenvalues are within the unit circle so the iterative solver is expected to converge in all the cases. Once this convergence is assured, MMS is used to verify a representative set of possible combinations introduced in Chapter 4. Again, no difference between the different finite element shapes is appreciated, as expected. In the case of non-conformal interfaces, the error is increased with respect to conformal interfaces due to the discontinuity

in the discretization between subdomains but the representation of the solution is not compromised.

Finally, a number of real problems are introduced in the last block, corresponding to Section 5.4. Apart from the validation of DDM with different electromagnetic problems and the expected performance already shown in the literature, a detailed study of the error provided by the non-conformal interface is introduced. This study plays a key role to assess the introduction of adaptivity in DDM since subdomains are refined independently non-conformal meshes on the interfaces are quite likely to appear. If different aspect ratios are tested, the error in the $|s_{21}|$ is quite flat. This means that the approximation of the field is not sensitive to the aspect ratio on the interface, which is convenient for adaptive meshes. Also, if the number of discontinuities which wave propagation has to cross is taken into account, a linear increase is detected but the dynamic range is small. Finally, if finer meshes are applied uniformly on the interface, the error in the $|s_{21}|$ is also quite similar. In view of these results, it can be stated that: *i)* independent refinement of each subdomain should not compromise the solution in the whole problem due to the lack of sensitivity in the aspect ratio; and *ii)* care has to be taken when introducing the division into subdomains for the slight increase in the error detected; and *iii)* if a uniform refinement is applied to the non-conformal interface, the error is not decreased.

In the last block, associated to the Chapter 6, an h adaptivity is introduced with DDM problems. First, an adaptivity algorithm for triangular prisms developed in the first block is included. Regarding the adaptivity algorithm itself, an estimator based on [153] is adapted to DDM; different marking strategies are included obtaining similar performance through MMS; and conformation techniques to obtain admissible meshes have to be included.

First results with DDM are obtained through MMS, and the so-called ROT criterion is used to keep the conformality on the interface. Then, a WR-90 waveguide is taken as problem to be solved obtaining the following conclusions: *i)* the error given by the estimator is clearly reduced in the areas where new elements are introduced; *ii)* the estimator is tested in real problems comparing the error obtained with the error provided by the analytical solution; *iii)* when the division between subdomains is conformal, same refinement meshes are obtained; and *iv)* non-conformal interfaces introduce an additional source of error which can lead to re-

fine in the non-conformal interface. A solution to this last point may be to penalize the error obtained by the estimator in the elements close to the non-conformal interfaces. It is worth noting that the error of the S-parameters is reduced if an adaptive refinement on the discontinuity is introduced, which is a promising result that will be further studied in the future.

In view of these conclusions, a singularity is introduced in a WR-90 waveguide by means of an L-shaped domain which is divided into five subdomains. Different experiments are performed, and it is observed that the error in the scattering parameters is flat with the number of iterations when the penalization in the elements close to the non-conformal interfaces is applied and only the subdomain with the singularity is refined. A possible explanation of this performance is the high aspect ratio present on the non-conformal interface, which grows geometrically with the number of iterations. If the interfaces are forced to be conformal, the error obtained for the $|s_{21}|$ parameter shows a better performance than uniform meshes. Also, two meshes with different p (one with a given distribution from the beginning and the other with a non-uniform p refinement with the same estimator used for h adaptivity) are included. No improvement compared to the h adaptive mesh is obtained as it may be expected since no special techniques for p adaptivity have been introduced: they are included as a proof of concept to check if the right answer is provided. Finally, if the representation of the singularity is measured by means of the maximum value of the field at the singularity, the best performance is obtained with non-conformal interfaces as it might be expected.

To sum up, the three building blocks to merge *hp* adaptivity with non-conformal and non-overlapping DDM have been successfully introduced:

- Basis functions with non-uniform p refinement have been introduced for a number of different element shapes, which provide more flexibility to adapt the finite element shape to the geometry of the problem.
- A non-conformal and non-overlapping DDM has been introduced and validated, and a thorough study of the non-conformality has been included in the dissertation.
- An adaptivity algorithm particularized for triangular prisms has

been introduced.

And on top of that, details about the parallel implementation have been given. A proof of concept of 3D *hp* adaptivity has been introduced, while a detailed study of the effect of non-conformal interfaces has been included. As far as authors know, this effect has not been studied with detail in the literature.

7.2 Future lines

The work presented in this dissertation has introduced useful building blocks that are employed to construct a prototype which merges *hp* adaptivity with non-conformal, non-overlapping DDM. The main future line which arises from this work is the validation of the conclusions obtained in this dissertation in large scale scenarios.

From the first block, additional results about the phase error may be computed. First, a better experiment to assess the performance of unstructured over structured meshes is to compute the error obtained with different lengths of the waveguide replicating the same mesh in sections of one wavelength. As expected by the 2-D analysis included in the literature, the structured meshes should perform worse than unstructured meshes due to the accumulation of phase error. However, all the experiments presented in this dissertation are not true dispersion errors but numerical errors. To compute the true dispersion error, the projection of the solution to the space of basis functions should be computed to obtain the so-called best approximation error, which can be used together with the numerical error to obtain the pollution or dispersion error, [27]. the solution This dispersion error in three dimensions has not been reported in the literature and this study will be performed for all the shapes and orders present in Chapter 4 in the future [1].

Regarding DDM, the introduction of new transmission conditions shown in the literature will be addressed to improve the convergence of the iterative solver in the future, [58]. Moreover, for repetitive structures only the cell element might be stored in order to save memory in the execution, [87].

Finally, with respect to *hp* adaptivity, unstructured finite elements (tetrahedra) will be introduced in the adaptivity algorithm to alleviate the high aspect ratio in the non-conformal interface obtained with struc-

[1] The author wants to thank Prof. Jin-Fa Lee for his suggestions on this topic.

tured elements. Moreover, to address this last problem, introduction of support of the so-called hanging nodes might be introduced since the error may be reduced. Special techniques for *hp* refinement already introduced in 2D, [134, 135], might be applied to each local subdomain with DDM. Furthermore, to achieve a scalable *hp* adaptivity code, a smarter balance for the MPI processes will be introduced trying to take into consideration the times needed for each subdomain to be refined.

7.3 Research stays

The work developed in this dissertation has been done mainly in Group of radiofrequency, electromagnetism, microwaves & antennas (GREMA) group, at University Carlos III of Madrid, Leganés, Spain. The author has been fundedby two grants obtained on a competitive basis: from 2014 to 2015, with a predoctoral grant from University Carlos III of Madrid and, from 2015 to 2018, with FPU grant from Spanish ministry of Education. It is also worth noting the collaboration with two different groups:

- Computational Electromagnetics Laboratory, at University of Macau, Macau.
- ElectroScience Laboratory, at the Ohio State University, Columbus, Ohio.

during the following stays:

- Six weeks with a short stay at University of Macau hosted by prof. K.W. Tam, from June 2015 to July 2015. Funded by University Carlos III of Madrid with a scholarship obtained on a competitive basis.
- Three months as visiting scholar at the Ohio State University hosted by Prof. Jin-Fa Lee, from September 2016 to December 2016. Funded by Spanish Ministry of Education with a scholarship obtained on a competitive basis.
- Three months as visiting scholar at the Ohio State University hosted by Prof. Jin-Fa Lee, from August 2017 to December 2017. Funded by University Carlos III of Madrid with a scholarship obtained on a competitive basis.

7.4 Publications

During this Ph.D. dissertation, different contributions have been reported to the literature. The complete list is detailed below.

Journals (3):

- **A. Amor-Martín**, L. E. García-Castillo, and D. García-Doñoro, "Second-Order Nédélec Curl-Conforming Prismatic Element for Computational Electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 10, pp. 4384-4395, 2016.
- D. García-Doñoro, S. Ting, **A. Amor-Martín**, and L. E. García-Castillo, "Analysis of Planar Microwave Devices using Higher Order Curl-Conforming Triangular Prismatic Finite Elements," *Microwave and Optical Technology Letters*, vol. 58, no. 8, pp. 1794-1801, 2016.
- **A. Amor-Martín**, I. Martínez-Fernandez, and L. E. García-Castillo, "Posidonia: A Tool for HPC and Remote Scientific Simulations [EM Programmer's Notebook]," *IEEE Antennas and Propagation Magazine*, vol. 57, no. 6, pp. 166-177, 2015.

Conference contributions (14):

- **A. Amor-Martín**, L. E. García-Castillo, and D. García-Doñoro, "Towards a Scalable hp Adaptive Finite Element Code based on a Nonconformal Domain Decomposition Method," in *European Microwave Conference (EuMC), 2018*, EuMA, 2018.
- D. García-Doñoro, W. Mei, **A. Amor-Martín**, L. E. García-Castillo, and M. Salazar-Palma, "Electromagnetic Finite Element Solver for HPC Environments using Direct Substructuring Method," in *European Microwave Conference (EuMC), 2018*, EuMA, 2018.
- **A. Amor-Martín**, L. E. García-Castillo, and D. García-Doñoro, "Non-conformal Domain Decomposition Method supporting hp Discretizations," in *14th International Workshop on Finite Elements for Microwave Engineering*, IEEE, 2018.
- **A. Amor-Martín**, L. E. García-Castillo, and D. García-Doñoro, "Higher Order Finite Element Method based on a Non-conformal Domain Decomposition Method," in *Emerging Trends in Applied Mathematics and Mechanics 2018*, 2018.

- J. A. Belloch-Rodriguez, **A. Amor-Martín**, D. García-Doñoro, and L. E. García-Castillo, "Acceleration of a Mesh Truncation Technique for a Finite Element Electromagnetics Code," in *18th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2018*, 2018.

- D. García-Doñoro, **A. Amor-Martín**, and L. E. García-Castillo, "Higher-Order Finite Element Electromagnetics Code for HPC Environments," *Procedia Computer Science*, vol. 108, pp. 818-827, 2017.

- **A. Amor-Martín**, D. García-Doñoro, and L. E. García-Castillo, "A Finite Element Mesh Truncation Technique for Scattering and Radiation Problems in HPC Environments," in *Computing and Electromagnetics International Workshop (CEM), 2017*, pp. 33-34, IEEE, 2017.

- **A. Amor-Martín**, D. García-Doñoro, and L. E. García-Castillo, "Analysis of Dispersion Error of Higher-Order Curl-Conforming Prismatic Finite Element," in *Numerical Electromagnetic and Multiphysics Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO), 2017 IEEE MTT-S International Conference on*, pp. 203-205, IEEE, 2017.

- **A. Amor-Martín**, D. García-Doñoro, and L. E. García-Castillo, "On the Design of Higher-Order Curl-Conforming Finite Elements and its Assembly Features," in *Numerical Electromagnetic and Multiphysics Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO), 2017 IEEE MTT-S International Conference on*, pp. 200-202, IEEE, 2017.

- L. García-Castillo, I. Gomez-Revuelto, **A. Amor-Martín**, M. Los, and M. Paszynski, "Algorithm for simultaneous adaptation and time step iterations for the electromagnetic waves propagation and heating of the human head induced by cell phone," *Procedia Computer Science*, vol. 108, pp. 2448-2452, 2017.

- **A. Amor-Martín** and L. E. García-Castillo, "Second-Order Nédélec Curl-Conforming Prism for Finite Element Computations," in *13th International Workshop on Finite Elements for Microwave Engineering*, IEEE, 2016.

- D. García-Doñoro, S. Ting, **A. Amor-Martín**, L. García-Castillo, and M. Salazar-Palma, "Higher Order Finite Element Method Solver for the Analysis of Microwave Devices in Planar Technology," in *Microwave Conference (EuMC), 2016 46th European*, pp. 473-476, IEEE, 2016.

- **A. Amor-Martín**, I. Martínez-Fernandez, and L. E. García-Castillo, "Posidonia: A tool for HPC and Remote Scientific Simulations," in *13th International Workshop on Finite Elements for Microwave Engineering*, IEEE, 2016.

- D. García-Doñoro, **A. Amor-Martín**, L. García-Castillo, M. Salazar-Palma, and T. Sarkar, "HOFEM: Higher Order Finite Element Method Simulator for Antenna Analysis," in *Antenna Measurements & Applications (CAMA), 2016 IEEE Conference on*, pp. 1-4, IEEE, 2016.

BIBLIOGRAPHY

[1] M. Salazar-Palma, T. K. Sarkar, L. E. García-Castillo, T. Roy, and A. R. Djordjevic, *Iterative and Self-Adaptive Finite-Elements in Electromagnetic Modeling.* Norwood, MA: Artech House Publishers, Inc., 1998.

[2] A. Amor-Martin and L. E. García-Castillo, "Second-order Nédélec curl-conforming prismatic element for computational electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 64, pp. 4384–4395, Oct. 2016.

[3] F. Fuentes, B. Keith, L. Demkowicz, and S. Nagaraj, "Orientation embedded high order shape functions for the exact sequence elements of all shapes," *Computers & Mathematics with applications*, vol. 70, no. 4, pp. 353–458, 2015.

[4] I. Babuvška and W. C. Rheinboldt, "Error estimates for adaptive finite element computations," *SIAM Journal on Numerical Analysis*, vol. 15, no. 4, pp. 736–754, 1978.

[5] P. Hennig, M. Kästner, P. Morgenstern, and D. Peterseim, "Adaptive mesh refinement strategies in isogeometric analysis—a computational comparison," *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 424–448, 2017.

[6] W. Dörfler, "A convergent adaptive algorithm for Poisson's equation," *SIAM Journal on Numerical Analysis*, vol. 33, no. 3, pp. 1106–1124, 1996.

[7] P. Monk, *Finite Element Methods for Maxwell Equations*. Oxford University Press, 2003.

[8] J. M. Jin, *The Finite Element Method in Electromagnetics*. John Wiley & Sons, Inc., 1993.

[9] J. M. Jin and D. J. Ryley, *Finite Element Analysis of Antennas and Arrays*. Wiley-IEEE Press, 2009.

[10] R. F. Harrington, *Field Computation by Moment Methods*. IEEE Press, 1993.

[11] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas and Propagation Magazine*, vol. 35, pp. 7–12, June 1993.

[12] K. S. Kunz and R. J. Luebbers, *The Finite Difference Time Domain method for electromagnetics*. CRC press, 1993.

[13] D. Garcia-Donoro, L. E. García-Castillo, and S. W. Ting, "Verification process of finite-element method code for electromagnetics: Using the method of manufactured solutions," *IEEE Antennas and Propagation Magazine*, vol. 7, pp. 28–38, Apr. 2016.

[14] D. Garcia-Donoro, L. E. Garcia-Castillo, T. K. Sarkar, and Z. Yu, "A non-standard Schwarz domain decomposition method for finite element mesh truncation of infinite arrays," *IEEE Transactions on Antennas and Propagation*, 2018.

[15] R. Fernández-Recio, L. E. Garcia-Castillo, S. L. Romano, and I. Gómez-Revuelto, "Convergence study of a non-standard Schwarz domain decomposition method for finite element mesh truncation in electromagnetics," *Progress In Electromagnetics Research (PIER)*, vol. 120, pp. 439–457, 2011.

[16] R. Ribó, M. Pasenau, E. Escolano, J. Ronda, and L. González, "GiD reference manual," *CIMNE, Barcelona*, vol. 27, 1998.

[17] T. D. Blacker, S. J. Owen, M. L. Staten, W. R. Quadros, B. Hanks, B. W. Clark, R. J. Meyers, C. Ernst, K. Merkley, R. Morris, *et al.*, "CUBIT geometry and mesh generation toolkit 15.1 user documentation.," tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2016.

[18] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities," *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[19] J. P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of Computational Physics*, vol. 114, pp. 185–200, Oct. 1994.

[20] P. G. Ciarlet, *The Finite Element Methods for Elliptic Problems.* New York: North Holland, 1994.

[21] J. C. Nédélec, "Mixed finite elements in $\Re^3$," *Numerische Mathematik*, vol. 35, pp. 315–341, 1980.

[22] J. C. Nédélec, "A new family of mixed finite elements in $\Re^3$," *Numerische Mathematik*, vol. 50, pp. 57–81, 1986.

[23] P. C. Hansen, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, vol. 4. Siam, 2005.

[24] N. Marsic and C. Geuzaine, "Efficient finite element assembly of high order Whitney forms," 2014.

[25] L. E. García-Castillo and M. Salazar-Palma, "Second-order Nédélec tetrahedral element for computational electromagnetics," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields (John Wiley & Sons, Inc.)*, vol. 13, pp. 261–287, March-June 2000.

[26] F. Fuentes, B. Keith, L. F. Demkowicz, and S. Nagaraj, "Orientation embedded high order shape functions for the exact sequence elements of all shapes," *Computer and Mathematics with Applications*, vol. 70, pp. 353–458, Aug. 2015.

[27] L. Demkowicz, *Computing with hp Finite Elements. I. One- and Two-Dimensional Elliptic and Maxwell Problems.* Chapman & Hall/CRC Press, Taylor and Francis, 2007.

[28] L. E. García-Castillo, D. Pardo, and L. F. Demkowicz, "A two-dimensional self-adaptive *hp*-adaptive finite element method for the characterization of waveguide discontinuities. Part I: Waveguide theory and finite element formulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 4823–4852, Nov. 2007.

[29] L. E. García-Castillo, D. Pardo, I. Gómez-Revuelto, and L. F. Demkowicz, "A two-dimensional self-adaptive *hp*-adaptive finite element method for the characterization of waveguide discontinuities. Part II: Energy-norm based automatic *hp*-adaptivity," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 4823–4852, Nov. 2007.

[30] J.-Y. L'Excellent, *Multifrontal methods: parallelism, memory usage and numerical aspects.* PhD thesis, Ecole normale supérieure de lyon-ENS LYON, 2012.

[31] D. Garcia-Donoro, A. Amor-Martin, and L. E. Garcia-Castillo, "Higher-order finite element electromagnetics code for HPC environments," *Procedia Computer Science*, vol. 108, pp. 818–827, 2017.

[32] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, "MUMPS: a general purpose distributed memory sparse solver," in *International Workshop on Applied Parallel Computing*, pp. 121–130, Springer, 2000.

[33] O. Schenk and K. Gärtner, "Solving unsymmetric sparse systems of linear equations with PARDISO," *Future Generation Computer Systems*, vol. 20, no. 3, pp. 475–487, 2004.

[34] Y. Saad, *Iterative Methods for Sparse Linear Systems.* SIAM, 2nd ed., 2003.

[35] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, "Efficient management of parallelism in object-oriented numerical software libraries," in *Modern software tools for scientific computing*, pp. 163–202, Springer, 1997.

[36] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp,

D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, "PETSc users manual," Tech. Rep. ANL-95/11 - Revision 3.10, Argonne National Laboratory, 2018.

[37] D. García Doñoro, *A New Software Suite for Electromagnetics*. PhD thesis, Department of Signal Theory and Communications, Universidad Carlos III de Madrid, July 2014.

[38] P. J. Roache, *Verification and Validation in Computational Science and Engineering*. Alburquerque, New Mexico, USA: Hermosa Publishers, 1998.

[39] D. Garcia-Donoro, S. Ting, A. Amor-Martin, L. Garcia-Castillo, and M. Salazar-Palma, "Higher order finite element method solver for the analysis of microwave devices in planar technology," in *Microwave Conference (EuMC), 2016 46th European*, pp. 473–476, IEEE, 2016.

[40] D. Garcia-Donoro, A. Amor-Martin, L. Garcia-Castillo, M. Salazar-Palma, and T. Sarkar, "HOFEM: Higher order finite element method simulator for antenna analysis," in *Antenna Measurements & Applications (CAMA), 2016 IEEE Conference on*, pp. 1–4, IEEE, 2016.

[41] D. Garcia-Donoro, S. Ting, A. Amor-Martin, and L. E. Garcia-Castillo, "Analysis of planar microwave devices using higher order curl-conforming triangular prismatic finite elements," *Microwave and Optical Technology Letters*, vol. 58, no. 8, pp. 1794–1801, 2016.

[42] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, and A. Zdunek, *Computing with hp Finite Elements. II Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*. Chapman & Hall/CRC Press, Taylor and Francis, 2008.

[43] L. F. Demkowicz, *Encyclopedia of Computational Mechanics*, ch. "Finite Element Methods for Maxwell Equations". John Wiley & Sons, Inc., 2004.

[44] L. F. Demkowicz, P. Monk, L. Vardapetyan, and W. Rachowicz, "De Rham diagram for *hp* finite element spaces," *Computer and Mathematics with Applications*, vol. 39, no. 7-8, pp. 29–38, 2000.

[45] J. P. Webb, "Hierarchal vector basis functions of arbitrary order for triangular and tetrahedral finite elements," *IEEE Transactions on Antennas and Propagation*, vol. 47, pp. 1244–1253, Aug. 1999.

[46] T. V. Yioultsis and T. B. Tsiboukis, "Convergence-optimized, higher order vector finite elements for microwave simulations," *IEEE Microwave and Wireless Components Letters*, vol. 11, pp. 419–421, Oct. 2001.

[47] M. M. Ilíc and B. M. Notaroš, "Higher order hierarchical curved hexaedral vector finite elements for electromagnetic modeling," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, pp. 1026–1033, Mar. 2003.

[48] Z. Ren and N. Ida, "Solving 3D eddy current problems using second order nodal and edge elements," *IEEE Transactions on Magnetics*, vol. 36, pp. 746–749, July 2000.

[49] D. K. Sun, J. F. Lee, and Z. Csendes, "Construction of nearly orthogonal Nedelec bases for rapid convergence with multilevel preconditioned solvers," *SIAM Journal of Scientific Computing*, vol. 23, no. 4, pp. 1053–1076, 2003.

[50] L. E. García-Castillo and M. Salazar-Palma, "Second-order Nédélec tetrahedral element for computational electromagnetics," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields (John Wiley & Sons, Inc.)*, vol. 13, pp. 261–287, March-June 2000.

[51] L. E. García-Castillo, A. J. Ruiz-Genovés, I. Gómez-Revuelto, M. Salazar-Palma, and T. K. Sarkar, "Third-order Nédélec curl-conforming finite element," *IEEE Transactions on Magnetics*, vol. 38, pp. 2370–2372, Sept. 2002.

[52] A. Amor-Martin, L. E. Garcia-Castillo, and D. D. Garcia-Doñoro, "Second-order Nédélec curl-conforming prismatic element for computational electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 10, pp. 4384–4395, 2016.

[53] T. Ozdemyr and J. L. Volakis, "Triangular prisms for edge-based vector finite element analysis of conformal antennas," *IEEE Transactions on Antennas and Propagation*, vol. 45, pp. 788–797, May 1997.

[54] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases on prism elements," *IEEE Transactions on Antennas and Propagation*, vol. 46, pp. 442–450, Mar. 1998.

[55] D. I. Karatzidis, T. V. Yioultsis, and T. B. Tsiboukis, "General multiple order prism vector finite macroelements for planar microwave circuits and EBGs," *IEEE Transactions on Magnetics*, vol. 45, pp. 1056–1059, Mar. 2009.

[56] D. I. Karatzidis, T. V. Yioultsis, and T. B. Tsiboukis, "Efficient analysis of planar microwave circuits with mixed-order prism vector finite macroelements," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields (John Wiley & Sons, Inc.)*, vol. 21, pp. 475–492, Nov. 2008.

[57] L. E. Tobon, Q. Ren, and Q. H. Liu, "Spectral-prism element for multi-scale layered package-chip co-simulations using the discontinuous galerkin time-domain method," *Electromagnetics*, vol. 34, pp. 270–285, 2014.

[58] V. Dolean, M. J. Gander, S. Lanteri, J.-F. Lee, and Z. Peng, "Effective transmission conditions for domain decomposition methods applied to the time-harmonic curl–curl Maxwell's equations," *Journal of computational physics*, vol. 280, pp. 232–247, 2015.

[59] M. F. Xue and J. M. Jin, "Nonconformal FETI-DP Methods for Large-scale Electromagnetic Simulation," *IEEE Transactions on Antennas and Propagation*, vol. 60, pp. 4291–4305, Sept 2012.

[60] B. Després, "A domain decomposition method for the harmonic Maxwell equations," *Iterative methods in linear algebra*, pp. 475–484, 1992.

[61] A. A. Rodriguez and L. Gerardo-Giorda, "New nonoverlapping domain decomposition methods for the harmonic Maxwell system," *SIAM Journal on Scientific Computing*, vol. 28, no. 1, pp. 102–122, 2006.

[62] V. Dolean, P. Jolivet, and F. Nataf, *An introduction to domain decomposition methods: algorithms, theory, and parallel implementation*, vol. 144. SIAM, 2015.

[63] F. Hecht, "New development in FreeFem++," *J. Numer. Math.*, vol. 20, no. 3-4, pp. 251–265, 2012.

[64] B. Thierry, A. Vion, S. Tournier, M. El Bouajaji, D. Colignon, N. Marsic, X. Antoine, and C. Geuzaine, "GetDDM: an open framework for testing optimized Schwarz methods for time-harmonic wave problems," *Computer Physics Communications*, vol. 203, pp. 309–330, 2016.

[65] A. St-Cyr, M. J. Gander, and S. J. Thomas, "Optimized multiplicative, additive, and restricted additive Schwarz preconditioning," *SIAM Journal on Scientific Computing*, vol. 29, no. 6, pp. 2402–2425, 2007.

[66] V. Dolean, M. J. Gander, and L. Gerardo-Giorda, "Optimized Schwarz methods for Maxwell's equations," *SIAM Journal on Scientific Computing*, vol. 31, no. 3, pp. 2193–2213, 2009.

[67] I. A. Baratta and E. J. Silva, "Multi-domain transmission conditions for domain decomposition methods applied to scattering problems," *IEEE Transactions on Magnetics*, vol. 54, no. 3, pp. 1–4, 2018.

[68] M. Bonazzoli, V. Dolean, F. Hecht, and F. Rapetti, "An example of explicit implementation strategy and preconditioning for the high order edge finite elements applied to the time-harmonic Maxwell's equations," *Computers & Mathematics with Applications*, vol. 75, no. 5, pp. 1498–1514, 2018.

[69] M. Bonazzoli, *Efficient high order and domain decomposition methods for the time-harmonic Maxwell's equations*. PhD thesis, Université Côte d'Azur, 2017.

[70] M. Bonazzoli, V. Dolean, F. Rapetti, and P.-H. Tournier, "Parallel preconditioners for high-order discretizations arising from full system modeling for brain microwave imaging," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 31, no. 2, p. e2229, 2018.

[71] M. E. Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri, "Optimized Schwarz methods for the time-harmonic Maxwell equations with damping," *SIAM Journal on Scientific Computing*, vol. 34, no. 4, pp. A2048–A2071, 2012.

[72] M. El Bouajaji, X. Antoine, and C. Geuzaine, "Approximate local magnetic-to-electric surface operators for time-harmonic Maxwell's equations," *Journal of Computational Physics*, vol. 279, pp. 241–260, 2014.

[73] M. El Bouajaji, B. Thierry, X. Antoine, and C. Geuzaine, "A quasi-optimal domain decomposition algorithm for the time-harmonic Maxwell's equations," *Journal of Computational Physics*, vol. 294, pp. 38–57, 2015.

[74] X. Antoine, "Fast approximate computation of a time-harmonic scattered field using the on-surface radiation condition method," *IMA Journal of Applied Mathematics*, vol. 66, no. 1, pp. 83–110, 2001.

[75] M. E. Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri, "Optimized Schwarz methods for the time-harmonic Maxwell equations with damping," *SIAM Journal on Scientific Computing*, vol. 34, no. 4, pp. A2048–A2071, 2012.

[76] N. Marsic, *Efficient methods for large-scale time-harmonic wave simulations*. PhD thesis, Université de Liége, Liége, Belgique, 2016.

[77] N. Marsic, C. Waltz, J.-F. Lee, and C. Geuzaine, "Domain decomposition methods for time-harmonic electromagnetic waves with high-order Whitney forms," *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 1–4, 2016.

[78] B. Stupfel and M. Lecouvez, "One-way domain decomposition method with exact radiation condition and fast GMRES solver for the solution of Maxwell's equations," *Journal of Computational Physics*, vol. 322, pp. 882–904, 2016.

[79] F. Ben Belgacem, A. Buffa, and Y. Maday, "The mortar finite element method for 3d Maxwell equations: first results," *SIAM Journal on Numerical Analysis*, vol. 39, no. 3, pp. 880–901, 2001.

[80] C. Bernardi, Y. Maday, and A. T. Patera, "A new non conforming approach to domain decomposition: The Mortar element method," in *Nonlinear Partial Differential Equations and Their Applications* (H. Brezis and J. L. Lions, eds.), pp. 13–51, Pitman, 1994.

[81] C. Bernardi, "A new nonconforming approach to domain decomposition: the mortar element method," *Nonlinear partial equations and their applications*, 1989.

[82] Y. Achdou, C. Japhet, Y. Maday, and F. Nataf, "A new cement to glue nonconforming grids with Robin interface conditions: The finite volume case," *Numerische Mathematik*, vol. 92, pp. 593–620, 2002.

[83] Z. Peng and J.-F. Lee, "Non-conformal domain decomposition method with second-order transmission conditions for time-harmonic electromagnetics," *Journal of Computational Physics*, vol. 229, no. 16, pp. 5615–5629, 2010.

[84] Z. Peng and J.-F. Lee, "A scalable nonoverlapping and nonconformal domain decomposition method for solving time-harmonic Maxwell equations in $\Re^3$," *SIAM Journal on Scientific Computing*, vol. 34, no. 3, pp. A1266–A1295, 2012.

[85] S.-C. Lee, M. N. Vouvakis, and J. F. Lee, "A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays," *Journal of Computational Physics*, vol. 203, pp. 1–21, Feb. 2005.

[86] M. N. Vouvakis, *A non-conformal domain decomposition method for solving large electromagnetic wave problems.* PhD thesis, The Ohio State University, 2005.

[87] M. N. Vouvakis, Z. Cendes, and J.-F. Lee, "A FEM domain decomposition method for photonic and electromagnetic band gap structures," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 2, pp. 721–733, 2006.

[88] M. Vouvakis, K. Zhao, S.-M. Seo, and J.-F. Lee, "A domain decomposition approach for non-conformal couplings between finite and boundary elements for unbounded electromagnetic problems in $\Re^3$," *Journal of Computational Physics*, vol. 225, no. 1, pp. 975–994, 2007.

[89] Z.-Q. Lu, X. An, and W. Hong, "A fast domain decomposition method for solving three-dimensional large-scale electromagnetic problems," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2200–2210, 2008.

[90] W.-J. Wang, R. Xu, H.-Y. Li, Y. Liu, X.-Y. Guo, Y. Xu, H.-L. Li, H.-J. Zhou, and W.-Y. Yin, "Massively parallel simulation of large-scale electromagnetic problems using one high-performance computing scheme and domain decomposition method," *IEEE Transactions on Electromagnetic Compatibility*, vol. 59, no. 5, pp. 1523–1531, 2017.

[91] V. Rawat, *Finite Element Domain Decomposition with Second Order Transmission Conditions for Time-Harmonic Electromagnetic Problems*. PhD thesis, The Ohio State University, 2009.

[92] K. Zhao, V. Rawat, S.-C. Lee, and J.-F. Lee, "A domain decomposition method with nonconformal meshes for finite periodic and semi-periodic structures," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 9, pp. 2559–2570, 2007.

[93] V. Rawat and J.-F. Lee, "Treatment of cement variables in the domain decomposition method for Maxwell's equations," in *Antennas and Propagation Society International Symposium, 2007 IEEE*, pp. 5937–5940, IEEE, 2007.

[94] V. Rawat and J.-F. Lee, "Nonoverlapping domain decomposition with second order transmission condition for the time-harmonic Maxwell's equations," *SIAM Journal on Scientific Computing*, vol. 32, no. 6, pp. 3584–3603, 2010.

[95] Z. Peng, V. Rawat, and J.-F. Lee, "One way domain decomposition method with second order transmission conditions for solving electromagnetic wave problems," *Journal of Computational Physics*, vol. 229, no. 4, pp. 1181–1197, 2010.

[96] Z. Peng and J.-F. Lee, "Non-conformal domain decomposition method with mixed true second order transmission condition for solving large finite antenna arrays," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 5, pp. 1638–1651, 2011.

[97] Y. Shao, Z. Peng, K. H. Lim, and J.-F. Lee, "Non-conformal domain decomposition methods for time-harmonic Maxwell equations," *Proc. R. Soc. A*, p. rspa20120028, 2012.

[98] J. Lu and J.-F. Lee, "An embedded domain decomposition method for time-harmonic electromagnetic problems," in *2018*

IEEE/MTT-S International Microwave Symposium-IMS, pp. 216–219, IEEE, 2018.

[99] D. Li, Y. Chen, C. Gunes, and J.-F. Lee, "Reverse operation self-consistent evaluation for curved domain interfaces in non-conformal domain decomposition methods," in *Antennas and Propagation (APSURSI), 2016 IEEE International Symposium on*, pp. 241–242, IEEE, 2016.

[100] M. Jiang, Y. Chen, D. Li, H. Zhao, S. Sun, Z. Nie, and J. Hu, "A flexible SIE-DDM for EM scattering by large and multiscale problems," *IEEE Transactions on Antennas and Propagation*, 2018.

[101] Z. Peng, X.-c. Wang, and J.-F. Lee, "Integral equation based domain decomposition method for solving electromagnetic wave scattering from non-penetrable objects," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 9, pp. 3328–3338, 2011.

[102] X. Wang, *A domain decomposition method for analysis of three-dimensional large-scale electromagnetic compatibility problems*. PhD thesis, The Ohio State University, 2012.

[103] Z. Peng, K.-H. Lim, and J.-F. Lee, "Nonconformal domain decomposition methods for solving large multiscale electromagnetic scattering problems," *Proceedings of the IEEE*, vol. 101, no. 2, pp. 298–319, 2013.

[104] H.-W. Gao, Z. Peng, and X.-Q. Sheng, "A geometry-aware domain decomposition preconditioning for hybrid finite element-boundary integral method," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 4, pp. 1875–1885, 2017.

[105] J. Moshfegh and M. Vouvakis, "Direct solution *of* FEM models: Are sparse direct solvers the best strategy?," in *Electromagnetics in Advanced Applications (ICEAA), 2017 International Conference on*, pp. 1636–1638, IEEE, 2017.

[106] J. Moshfegh and M. N. Vouvakis, "A memory-efficient sparse direct solver with applications in CEM," in *Antennas and Propagation & USNC/URSI National Radio Science Meeting, 2017 IEEE International Symposium on*, pp. 1577–1578, IEEE, 2017.

[107] M. N. Vouvakis and J. Moshfegh, "Sparse direct matrix solvers of finite element discretizations in electromagnetics," in *Applied Computational Electromagnetics Society Symposium (ACES), 2018 International*, pp. 1–2, IEEE, 2018.

[108] M. Xue, *Unified conformal/nonconformal domain decomposition methods for solving large-scale multi-region electromagnetic problems.* PhD thesis, 2014.

[109] Y. Li and J.-M. Jin, "A vector dual-primal finite element tearing and interconnecting method for solving 3–D large-scale electromagnetic problems," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 10, pp. 3000–3009, 2006.

[110] Y.-J. Li and J.-M. Jin, "Parallel implementation of the FETI-DPEM algorithm for general 3d em simulations," *Journal of Computational Physics*, vol. 228, no. 9, pp. 3255–3267, 2009.

[111] M.-F. Xue and J.-M. Jin, "Finite-element domain decomposition methods for analysis of large-scale electromagnetic problems.," *Applied Computational Electromagnetics Society Journal*, vol. 29, no. 12, 2014.

[112] M.-F. Xue and J.-M. Jin, "A preconditioned dual–primal finite element tearing and interconnecting method for solving three-dimensional time-harmonic Maxwell's equations," *Journal of Computational Physics*, vol. 274, pp. 920–935, 2014.

[113] K. Zhang and J.-M. Jin, "Parallel FETI-DP algorithm for efficient simulation of large-scale em problems," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 29, no. 5, pp. 897–914, 2016.

[114] I. Voznyuk, *Domain decomposition method for electromagnetic modelling and quantitative microwave imaging in large-scale three-dimensional configurations.* PhD thesis, Aix-Marseille Université, 2014.

[115] I. Voznyuk, H. Tortel, and A. Litman, "Scattered field computation with an extended FETI-DPEM2 method," *Progress In Electromagnetics Research*, vol. 139, pp. 247–263, 2013.

[116] I. Voznyuk, H. Tortel, and A. Litman, "3-D electromagnetic scattering computation in free-space with the FETI-FDP2 method," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 6, pp. 2604–2613, 2015.

[117] J. Ma, J.-M. Jin, and Z. Nie, "A nonconformal FEM-DDM with tree–cotree splitting and improved transmission condition for modeling subsurface detection problems," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 355–364, 2014.

[118] J. Ma and Z.-P. Nie, "FEM-DDM with an efficient second-order transmission condition in both high-frequency and low-frequency applications," *Progress In Electromagnetics Research*, vol. 50, pp. 253–271, 2013.

[119] T. Wan and Z. Jiang, "Multilevel compressed block decomposition–based finite-element domain decomposition method for the fast analysis of finite periodic structures," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 30, no. 5, p. e2194, 2017.

[120] F.-X. Roux and A. Barka, "Block krylov recycling algorithms for FETI-2LM applied to 3-D electromagnetic wave scattering and radiation," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 4, pp. 1886–1895, 2017.

[121] M. M. Botha and D. B. Davidson, "An explicit a posteriori error indicator for electromagnetic, finite element-boundary integral analysis," *IEEE Transactions on antennas and propagation*, vol. 53, no. 11, pp. 3717–3725, 2005.

[122] J. M. Melenk and B. I. Wohlmuth, "On residual-based a posteriori error estimation in *hp*-FEM," *Advances in Computational Mathematics*, vol. 15, no. 1-4, pp. 311–331, 2001.

[123] M. Bürg, "A residual-based a posteriori error estimator for the hp-finite element method for Maxwell's equations," *Applied Numerical Mathematics*, vol. 62, no. 8, pp. 922–940, 2012.

[124] D. Boffi, L. Gastaldi, R. Rodríguez, and I. Šebestová, "Residual-based a posteriori error estimation for the Maxwell's eigenvalue problem," *IMA Journal of Numerical Analysis*, vol. 37, no. 4, pp. 1710–1732, 2017.

[125] R. E. Bank, A. H. Sherman, and A. Weiser, "Some refinement algorithms and data structures for regular local mesh refinement," *Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences*, vol. 1, pp. 3–17, 1983.

[126] R. Verfürth, "A posteriori error estimation and adaptive mesh-refinement techniques," *Journal of Computational and Applied Mathematics*, vol. 50, no. 1-3, pp. 67–83, 1994.

[127] S. Chalasani and D. Thompson, "Quality improvements in extruded meshes using topologically adaptive generalized elements," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 6, pp. 1139–1159, 2004.

[128] G. Zboinskif, "Triangular-prism *hpg* adaptive finite element," *Computers & Structures*, vol. 65, no. 4, pp. 497–514, 1997.

[129] O. Meister and M. Bader, "2D adaptivity for 3D problems: Parallel SPE10 reservoir simulation on dynamically adaptive prism grids," *Journal of computational science*, vol. 9, pp. 101–106, 2015.

[130] W. Dörfler and V. Heuveline, "Convergence of an adaptive *hp* finite element strategy in one space dimension," *Applied numerical mathematics*, vol. 57, no. 10, pp. 1108–1124, 2007.

[131] C. Kreuzer and K. G. Siebert, "Decay rates of adaptive finite elements with Dörfler marking," *Numerische Mathematik*, vol. 117, no. 4, pp. 679–716, 2011.

[132] M. Schober and M. Kasper, "Comparison of *hp*-adaptive methods in finite element electromagnetic wave propagation," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 26, no. 2, pp. 431–446, 2007.

[133] W. F. Mitchell and M. A. McClain, "A survey of *hp*-adaptive strategies for elliptic partial differential equations," in *Recent advances in computational and applied mathematics*, pp. 227–258, Springer, 2011.

[134] L. E. García-Castillo, D. Pardo, I. Gómez-Revuelto, and L. F. Demkowicz, "A two-dimensional self-adaptive *hp* finite element

method for the characterization of waveguide discontinuities. Part I: Energy-norm based automatic *hp*-adaptivity," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 4823–4852, Nov. 2007. doi:10.1016/j.cma.2007.06.024.

[135] L. E. García-Castillo, D. Pardo, L. F. Demkowicz, and C. Torres-Verdín, "A two-dimensional self-adaptive *hp* finite element method for the characterization of waveguide discontinuities. Part II: Goal-oriented *hp*-adaptivity," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 4811–4822, Nov. 2007. doi:10.1016/j.cma.2007.06.023.

[136] I. Gomez-Revuelto, L. E. Garcia-Castillo, S. Llorente-Romano, and D. Pardo, "A three-dimensional self-adaptive *hp* finite element method for the characterization of waveguide discontinuities," *Computer Methods in Applied Mechanics and Engineering*, no. 249–252, pp. 62–74, 2012.

[137] J. W. Arthur, "The evolution of Maxwell's equations from 1862 to the present day," *IEEE Antennas and Propagation Magazine*, vol. 55, no. 3, pp. 61–81, 2013.

[138] "MUMPS Solver." `http://www.enseeiht.fr/lima/apo/MUMPS/`.

[139] *Intel Math Kernel Library. Reference Manual.* Intel Corporation, 2009.

[140] "METIS: Serial graph partitioning and fill-reducing matrix ordering." `http://glaros.dtc.umn.edu/gkhome/metis/metis/overview`.

[141] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, "PETSc Web page." `http://www.mcs.anl.gov/petsc`, 2017.

[142] J. Liu and J. M. Jin, "Scattering analysis of large body with deep cavities," *IEEE Transactions on Antennas and Propagation*, vol. 51, pp. 1157–1167, June 2003.

[143] P. G. Ciarlet, "The finite element method for elliptic problems," *Classics in applied mathematics*, vol. 40, pp. 1–511, 2002.

[144] J. T. Oden and J. N. Reddy, *An Introduction to the Mathematical Theory of Finite Elements.* Pure and Applied Mathematics. A Wiley-Interscience Series of Texts, Monographs and Tracts, John Wiley & Sons, Inc., 1976.

[145] J.-Y. Wu and R. Lee, "The advantages of triangular and tetrahedral edge elements for electromagnetic modeling with the finite-element method," *IEEE Transactions on Antennas and Propagation*, vol. 45, pp. 1431–1437, Sep 1997.

[146] F. Ihlenburg, *Finite element analysis of acoustic scattering*, vol. 132. Springer Science & Business Media, 2006.

[147] "GiD: The personal pre and postprocessor." International Center for Numerical Methods in Engineering (CIMNE), URL. `URL:http://gid.cimne.upc.es/`.

[148] D. García Doñoro, "A new software suite for electromagnetics," July 2014. Tesis Doctoral. Universidad Carlos III de Madrid. Calificación: Sobresaliente *cum laude*.

[149] H. M. El Misilmani, M. Al-Husseini, and K. Y. Kabalan, "Design of slotted waveguide antennas with low sidelobes for high power microwave applications," *Progress In Electromagnetics Research*, vol. 56, pp. 15–28, 2015.

[150] F. Alessandri, M. Chiodetti, A. G. D. Maiarelli, G. Martirano, D. Schmitt, L. Vanni, and F. Vitulli, "The electric-field integral-equation method for the analysis and design of a class of rectangular cavity filters loaded by dielectric and metallic cylindrical pucks," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, pp. 1790–1797, Aug. 2004.

[151] H. T. Nguyen, *p-adaptive and automatic hp-adaptive finite element methods for elliptic partial differential equations.* PhD thesis, UC San Diego, 2010.

[152] A. Wexler, "Solution of waveguide discontinuities by modal analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 15, no. 9, pp. 508–517, 1967.

[153] M. M. Botha and D. B. Davidson, "An explicit a *Posteriori* error indicator for electromagnetic, finite element-boundary integral analysis," *IEEE Transactions on Antennas and Propagation*, vol. 53, pp. 3717–3724, Nov. 2005.

About the Author

Adrián Amor-Martín was born in Móstoles, Spain, on May 17, 1989. He received the Engineer degree in Telecommunications and the Master degree in Multimedia and Communications from University Carlos III of Madrid in 2012 and 2014, respectively. Since 2012 he is working with the Group of Radiofrequency, Electromagnetism and Antennas at University Carlos III of Madrid. His research interests are focused on the application of numerical methods to high-performance computational electromagnetics including finite elements, domain decomposition methods, and *hp* adaptivity.